บทที่ 5
แบบจำลองความสัมพันธ์ระหว่างเอนทิตี

แบบจำลองความสัมพันธ์ระหว่างเอนทิตีเป็นแบบจำลองเชิงแนวคิด โดยนำเสนอในแบบจำลอง E-R Diagram หรือ E-R จุดประสงค์ของแบบจำลองเชิงแนวคิดคือ การนำเสนอให้เกิดความเข้าใจระหว่างผู้ออกแบบและผู้ใช้งาน นักออกแบบฐานข้อมูลจะใช้แบบจำลองเชิงแนวคิดสำหรับการสื่อสารกับผู้ใช้เพื่อให้เกิดความเข้าใจตรงกันเกี่ยวกับข้อมูลที่เกี่ยวข้องในระบบ

ในบทที่ 3 ได้กล่าวถึงแบบจำลองฐานข้อมูลชนิดต่างๆ ซึ่งในบทนี้จะกล่าวถึงแบบจำลองฐานข้อมูลเชิงสัมพันธ์เป็นหลัก เครื่องมือที่ใช้สำหรับการออกแบบฐานข้อมูลเชิงแนวคิดคือ แบบจำลอง Entity-Relationship Diagram (E-R Diagram) หรือ E-R ทั้งนี้ แบบจำลอง E-R เป็นแบบจำลองที่นำเสนอด้วยแผนภาพหรือไดอะแกรมที่ใช้สำหรับการสื่อสารให้เกิดความเข้าใจร่วมกันระหว่างผู้วิเคราะห์และผู้ใช้ระบบ โดยเสนอข้อมูลของระบบในภาพรวม ได้แก่ เอนทิตี แอตทริบิวต์ และประเภทของความสัมพันธ์ระหว่างเอนทิตี เป็นต้น ซึ่งผู้ใช้สามารถเข้าใจได้ง่าย และสามารถให้ข้อมูลต่อนักออกแบบฐานข้อมูลเพื่อนำไปพัฒนาเป็นแบบจำลอง E-R ที่สมบูรณ์ต่อไป (Sikha & Richard, 2012, p. 187)

แบบจำลองฐานข้อมูลเชิงสัมพันธ์

ปี ค.ศ 1976 Peter Chen ได้พัฒนาแบบจำลอง E-R ขึ้นมาใช้เพื่อการออกแบบฐานข้อมูล และนับจากนั้นเป็นต้นมา แบบจำลอง E-R ก็ได้มีการนำไปใช้อย่างกว้างขวาง มีการพัฒนารูปแบบที่หลากหลาย แต่รูปแบบของแบบจำลอง E-R อยู่บนพื้นฐานแนวความคิดเดียวกัน ในเบื้องต้นจะขอกล่าวถึงแบบจำลอง E-R แบบดั้งเดิมก่อน (Peter Chen Model) เนื่องจากมีการใช้งานอย่างแพร่หลายง่ายต่อการเรียนรู้และทำความเข้าใจ โดยหลังจากได้เรียนรู้แบบจำลอง E-R แบบดั้งเดิมเป็นที่เข้าใจแล้ว จึงค่อยนำเสนอแบบจำลอง E-R ในรูปแบบ Crow's Foot Model ต่อไป ในการที่สร้างแบบจำลอง E-R ได้นั้น จำเป็นต้องเรียนรู้ถึงสัญลักษณ์ที่ใช้แทนความหมายต่างๆ ให้เข้าใจเสียก่อน ซึ่งประกอบด้วยรายละเอียดที่สำคัญดังต่อไปนี้ (Batini, 1992, p. 207)

1. เอนทิตี

เอนทิตี (entity) คือวัตถุที่เราสนใจ ซึ่งอาจเป็นได้ทั้งบุคคล สถานที่ วัตถุ เหตุการณ์ หรือแนวคิดที่ก่อให้เกิดกลุ่มของข้อมูลที่ต้องการ ตัวอย่างของเอนทิตีต่างๆ มีดังนี้

บุคคล :	EMPLOYEE, STUDENT, PATIENT (พนักงาน, นักศึกษา, ผู้ป่วย)
สถานที่ :	STORE, WAREHOUSE, STATE (ร้านค้า, คลังสินค้า, สถานี)
วัตถุ :	MACHINE, BUILDING, AUTOMOBILE (เครื่องจักร, ตึกอาคาร, เครื่องยนต์)
เหตุการณ์ :	SALE, REGISTRATION, RESERVATION (การขาย, การลงเบียน, การจองห้องพัก)
แนวคิด :	ACCOUNT, COURES, WORK CENTER (บัญชี, หลักสูตร, ศูนย์การทำงาน)

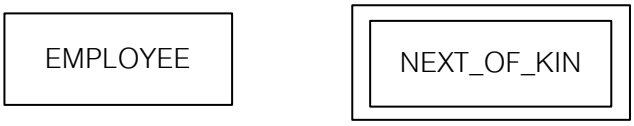
สัญลักษณ์เอนทิตีจะเป็นรูปสี่เหลี่ยมผืนผ้าดังภาพที่ 5.1 สำหรับการตั้งชื่อให้กับเอนทิตีนั้น ชื่อที่ใช้จะต้องเป็น **คำนาม** และโดยปกติจะเขียนด้วย **ตัวพิมใหญ่**

ภาพที่ 5.1 สัญลักษณ์เอนทิตี ซึ่งประกอบด้วยเอนทิตี EMPLOYEE และเอนทิตี BRANCH
ที่มา : Batini (1992, p. 207)

นอกจากนี้ เอนทิตียังสามารถแบ่งออกเป็น 2 ประเภทด้วยกันคือ

1.1 **เอนทิตีปกติ** (Strong Entity) คือเอนทิตีที่เกิดขึ้นได้ด้วยตัวเอง เป็นอิสระไม่ขึ้นกับเอนทิตีใดๆ สัญลักษณ์ที่ใช้คือรูปสี่เหลี่ยมผืนผ้า และสามารถเรียก strong entity ได้อีกชื่อหนึ่งว่า regular entity

1.2 **เอนทิตีอ่อนแอ** (Weak Entity) คือเอนทิตีที่ขึ้นอยู่กับเอนทิตีชนิดอื่น ๆ ไม่สามารถเกิดขึ้นได้ตามลำพัง และจะถูกลบเมื่อเอนทิตีหลักถูกลบออกไป สัญลักษณ์ที่ใช้คือรูปสี่เหลี่ยมผืนผ้าเช่นกันแต่เป็นเส้นคู่



ภาพที่ 5.2 เอนทิตีปกติและเอนทิตีอ่อนแอ
ที่มา : Batini (1992, p. 208)

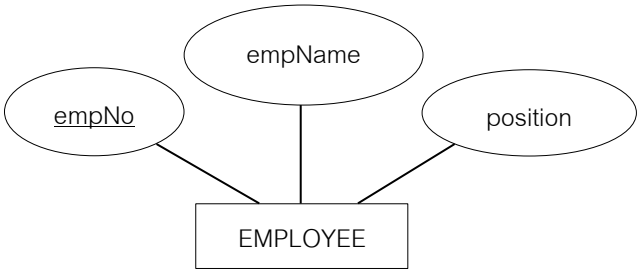
จากภาพที่ 5.2 เอนทิตี EMPLOYEE (พนักงาน) เป็นเอนทิตีปกติ ในขณะที่ NEXT_OF_KIN (ญาติ) เป็น เอนทิตีอ่อนแอ หมายความว่า พนักงานจะต้องระบุญาติใกล้ชิดที่สามารถติดต่อได้ ดังนั้นเอนทิตี NEXT_OF_KIN จึงขึ้นอยู่กับเอนทิตี EMPLOYEE เมื่อเอนทิตี EMPLOYEE ถูกลบออกไป เอนทิตี NET_OF_KIN ก็จะถูกลบออกตามไปด้วย

2. แอตทริบิวต์

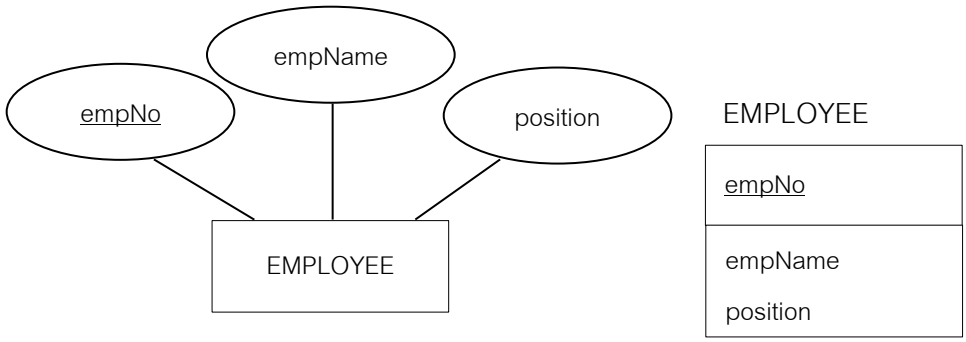
แอตทริบิวต์ (attributes) คือคุณสมบัติของเอนทิตี สัญลักษณ์แอตทริบิวต์จะเป็นรูปวงรี แอตทริบิวต์ใดที่ถูกใช้เป็นคีย์หลักจะถูกขีดเส้นใต้กำกับ พิจารณาจากภาพที่ 5.3 - 5.5 จะเห็นได้ว่าเอนทิตี EMPLOYEE ประกอบด้วย

แอตทริบิวต์ empNo, empName และ position โดยมีการกำหนดให้แอตทริบิวต์ empNo เป็นคีย์หลัก และหากนำไปเขียนในรูปแบบของรีเลชั่นสคีมา ก็สามารถเขียนได้ดังนี้

EMPLOYEE (empNo, empName, position)

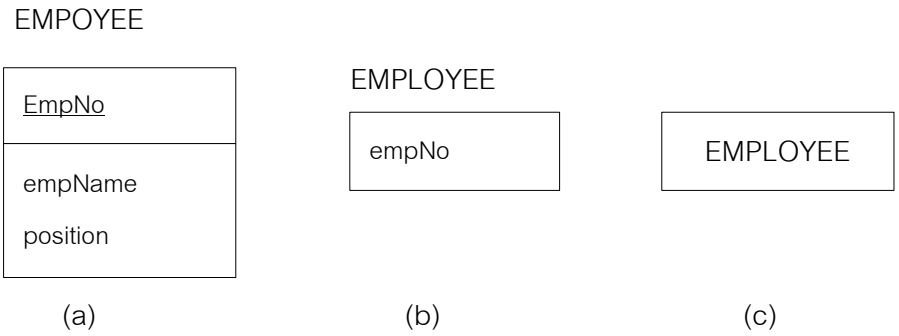


ภาพที่ 5.3 เอนทิตี EMPLOYEE ประกอบด้วยแอตทริบิวต์ต่างๆ โดยมีแอตทริบิวต์ empNo เป็นคีย์หลัก
ที่มา : Batini (1992, p. 208)



ภาพที่ 5.4 (a) แอตทริบิวต์ที่อยู่ภายในสัญลักษณ์วงรี
(b) แอตทริบิวต์ที่อยู่ในสัญลักษณ์สี่เหลี่ยม

ที่มา : Batini (1992, p. 209)



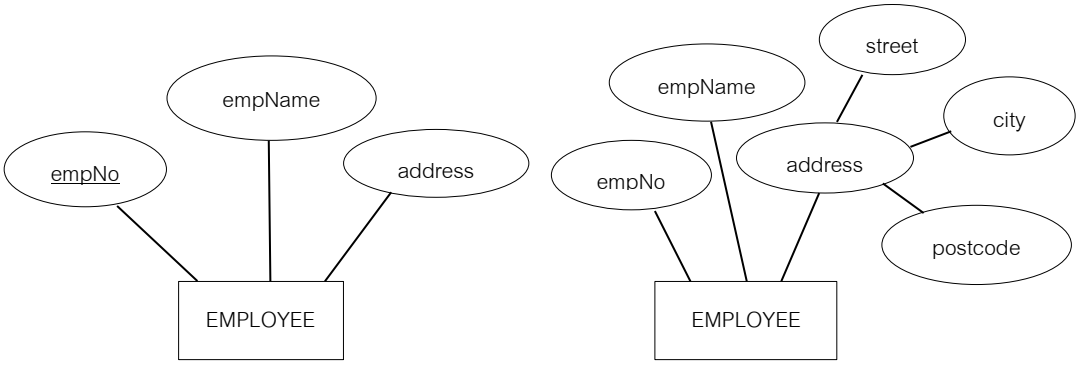
ภาพที่ 5.5 (a) เอนทิตีที่บรรจุแอตทริบิวต์ทุกตัว
(b) เอนทิตีที่ระบุเพียงแอตทริบิวต์ที่ใช้เป็นคีย์หลัก
(c) เอนทิตีที่ไม่แสดงแอตทริบิวต์ใดๆ

ที่มา : Batini (1992, p. 210)

แอตทริบิวต์สามารถแบ่งออกเป็นหลายชนิด ดังต่อไปนี้

2.1 **แอตทริบิวต์โดเมน** (Attribute Domain) เป็นการกำหนดขอบเขตค่าข้อมูลและชนิดของข้อมูลให้กับแอตทริบิวต์ โดยโดเมนกำหนดความเป็นไปได้ของข้อมูลในแอตทริบิวต์นั้นๆ ตัวอย่างเช่น attribute domain ของแอตทริบิวต์ sex จะมีค่าที่เป็นไปได้คือ M หรือ F ได้เท่านั้น เป็นต้น

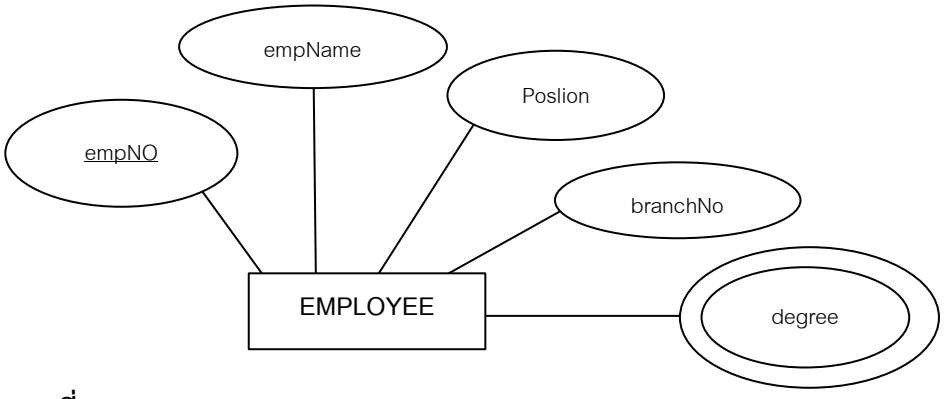
2.2 แอตทริบิวต์อย่างง่าย (Simple Attribute) และแอตทริบิวต์แบ่งย่อย (Composite Attribute) ความหมายของคำว่า Simple Attribute คือแอตทริบิวต์ที่มีองค์ประกอบเดียว กล่าวคือเป็นแอตทริบิวต์ที่ม่สามารถแตกเป็นข้อมูลย่อยได้อีก ตัวอย่างเช่น age, sex และ salary เป็นต้น ในบางครั้งเราอาจเรียก simple attribute ได้อีกชื่อหนึ่งว่า atomic attributes ส่วน composite attribute คือแอตทริบิวต์ที่มีองค์ประกอบอยู่หลายตัว มีข้อมูลมากกว่าหนึ่งค่าภายในแอตทริบิวต์นั้น เช่น แอตทริบิวต์ address ที่ภายในประกอบด้วยแอตทริบิวต์ street, city และ postcode ดังภาพที่ 5.6 (b)



(a) (b)

ภาพที่ 5.6 เปรียบเทียบระหว่าง Simple Attribute และ Composite Attribute
ที่มา : Batini (1992, p. 211)

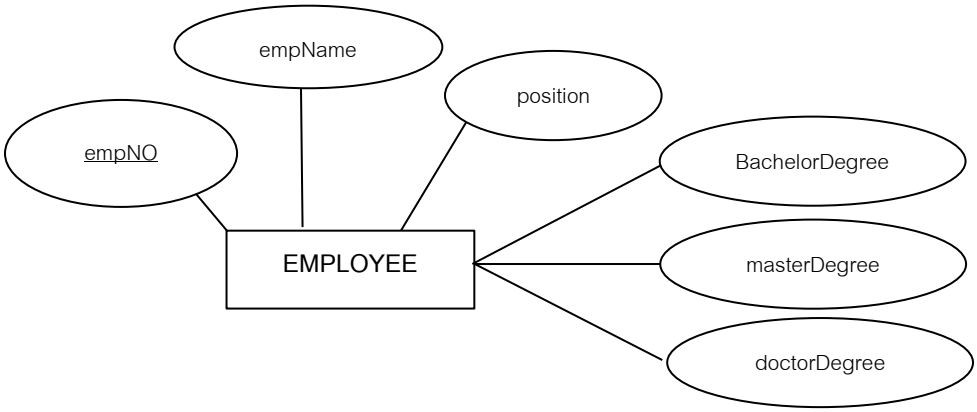
2.3 แอตทริบิวต์ที่มีค่าเดียว (Single-valued Attribute) และแอตทริบิวต์ที่มีหลายค่า (Multivalued Attribute) ความหมายของ single-valued attribute คือ แอตทริบิวต์ที่บรรจุค่าเพียงค่าเดียว ตัวอย่างเช่น พนักงานต้องทำงานอยู่ในสาขา ดังนั้น แอตทริบิวต์ branchNo ในเอนทิตี EMPLOYEE จึงสามารถนำไปอ้างอิงข้อมูลสาขาได้เพียงหนึ่งสาขาเท่านั้น แต่อย่างไรก็ตาม single-valued attribute ไม่จำเป็นต้องเป็น simple attribute เสมอไป ตัวอย่างเช่น รหัสสินค้าหมายเลข SE-08-02-189935 คือ Single-Valued Attribute แต่แท้จริงแล้วคือ composite attribute เพราะว่าสามารถแบ่งย่อยข้อมูลออกไปได้อีก โดยสมมติว่า SE แทนรหัสประกันสินค้า และ 189935 คือเลขรหัสสินค้า เป็นต้น ในขณะที่ multivalued attribute คือ แอตทริบิวต์ที่สามารถมีหลายค่าได้ ตัวอย่างเช่น พนักงานคนหนึ่งสามารถมีวุฒิการศึกษาได้หลายระดับ หรือภายในครอบครัวหนึ่งสามารถมีเบอร์โทรศัพท์ได้หลายเบอร์ เป็นต้น โดยสัญลักษณ์เส้นวงรีของ multivalued attribute จะเป็นเส้นคู่ ดังภาพที่ 5.7



ภาพที่ 5.7 Single-Value และ Multivalued Attribute
ที่มา : Batini (1992, p. 212)

แนวทางในการแก้ไขปัญหา multivalued attribute มีอยู่ 2 แนวทาง โดยในที่นี้จะใช้วุฒิการศึกษาเป็นตัวอย่าง ซึ่งแต่ละคนสามารถมีได้หลายวุฒิ

1. ที่เอนทิตีหลัก ให้ทำการสร้างแอตทริบิวต์เพิ่มเติมขึ้นมาใหม่ เช่น bachelorDegree ใช้แทน วุฒิปริญญาตรี, masterDegree ใช้แทนวุฒิปริญญาโท และ doctorDegree ใช้แทนวุฒิปริญญาเอก ซึ่งสามารถเขียนใหม่ได้ดังภาพที่ 5.8

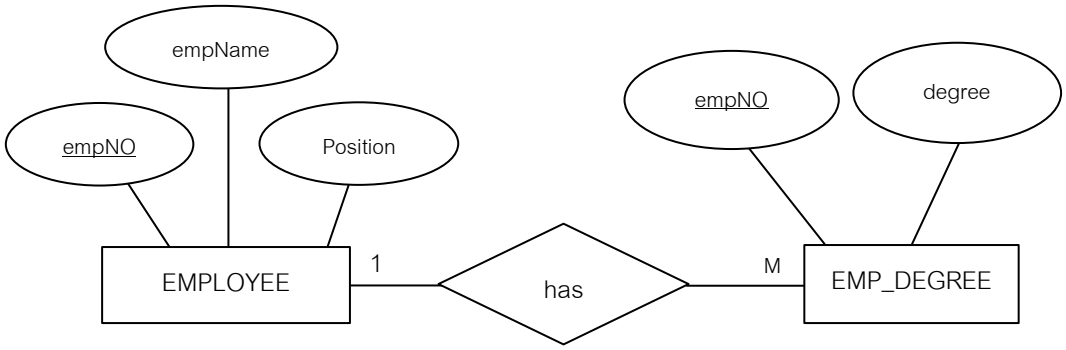


ภาพที่ 5.8 แสดงการแตก Multivalued Attribute ไปเป็นแอตทริบิวต์ใหม่
ที่มา : Batini (1992, p. 213)

อย่างไรก็ตาม เมื่อพิจารณาโดยคร่าวๆ แล้วดูเหมือนว่าเป็นแนวทางแก้ปัญหาที่ดี แต่หากพิจารณาในรายละเอียดแล้วจะพบว่า พนักงานบางคนอาจมีวุฒิการศึกษาระดับ

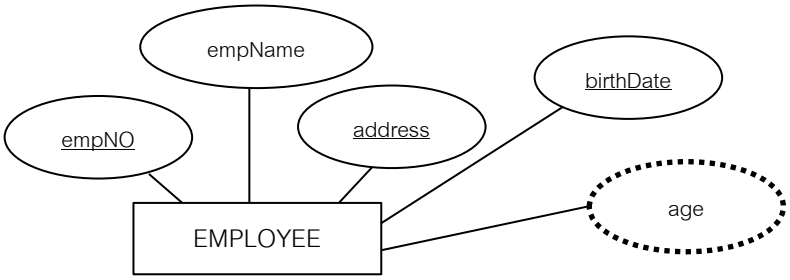
ปริญญาตรีหลายใบรวมถึงมีวุฒิปริญญาโทหรือวุฒิปริญญาเอกหลายใบก็เป็นได้ในขณะที่พนักงานบางคนอาจจบวุฒิปริญญาตรีเพียงวุฒิเดียว ดังนั้นแอตทริบิวต์วุฒิการศึกษาอื่นๆ ก็จะมีค่าเป็น Null ทำให้สิ้นเปลืองเนื้อที่โดยใช่เหตุ

2. ดำเนินการสร้างเอนทิตีใหม่ขึ้นมา ในที่นี้คือ EMP_DEGREE โดยกำหนดให้มีความสัมพันธ์แบบ one-to-many ซึ่งเป็นไปดังภาพที่ 5.9 ดังนี้คือ พนักงานสามารถมีวุฒิการศึกษาต่างๆ ได้หลายระดับ กล่าวคือ พนักงานแต่ละคนจะมีวุฒิการศึกษาได้หลายระดับหรือหลายใบตามความเป็นจริงของแต่ละบุคคล



ภาพที่ 5.9 แสดงการแตก Multivalued Attribute ไปเป็นเอนทิตีใหม่
ที่มา : Batini (1992, p. 214)

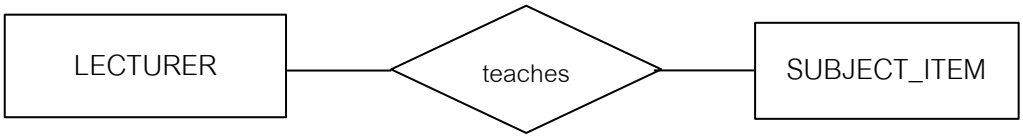
2.4 แอตทริบิวต์ไม่ต้องจัดเก็บ (Derived Attribute) คือ แอตทริบิวต์ที่ได้มาจากการประยุกต์ด้วยแอตทริบิวต์ เช่น แอตทริบิวต์ age ได้จากการนำแอตทริบิวต์วันเกิด (birthDate) มาประยุกต์ใช้งานด้วยการนำปีปัจจุบันหักด้วยปีเกิดจะได้แอตทริบิวต์ age โดยสัญลักษณ์ของแอตทริบิวต์ชนิดนี้จะเป็นรูปวงรีแบบเส้นประสำหรับ derived attribute ยังสามารถเรียกอีกชื่อหนึ่งว่า computed attribute ดังภาพที่ 5.10



ภาพที่ 5.10 Derived Attribute
ที่มา : Batini (1992, p. 214)

ความสัมพันธ์ระหว่างเอนทิตี

ความสัมพันธ์ระหว่างเอนทิตี (relationship) ควรมีชื่อระบุไว้เพื่อใช้อธิบาย ซึ่งปกติจะใช้สัญลักษณ์สี่เหลี่ยมขนมเปียกปูน (◇) ที่ภายในระบุคำกริยาไว้เพื่ออธิบายความสัมพันธ์ ตัวอย่างเช่น อาจารย์ <สอน> หนังสือได้หลายรายวิชา เป็นต้น (Connolly & Begg, 2005, p. 346-350)

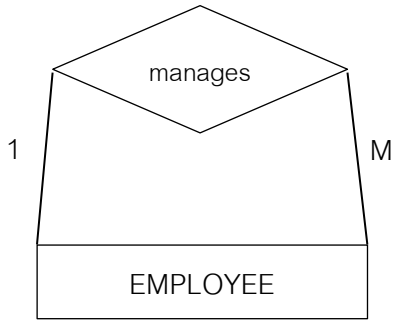


ภาพที่ 5.11 แสดงการอธิบายความสัมพันธ์ระหว่างอาจารย์และรายวิชาที่สอน
ที่มา : Connolly & Begg (2005, p. 346)

1. ดีกรีของความสัมพันธ์

ดีกรีของความสัมพันธ์ (relationship degree) เป็นตัวแสดงถึงจำนวนเอนทิตีที่มีความสัมพันธ์ระหว่างกัน โดยพื้นฐานความเป็นไปได้ของจำนวนความสัมพันธ์ระหว่างเอนทิตีมีอยู่อยู่ 3 รูปแบบด้วยกันคือ

1.1 **ความสัมพันธ์แบบยูนารี (Unary Relationships)** เป็นความสัมพันธ์ระหว่างเอนทิตีเดียว ตัวอย่างเช่น ผู้จัดการ *<จัดการ>* พนักงานของตน (ผู้จัดการก็คือพนักงาน) ซึ่งแสดงไว้ดังภาพที่ 5.12 โดยสัมพันธ์แบบยูนารีนี้ก็คือ ความสัมพันธ์แบบรีเคอร์ซีพ (recursive) ที่เรียกใช้ตัวเองนั่นเอง



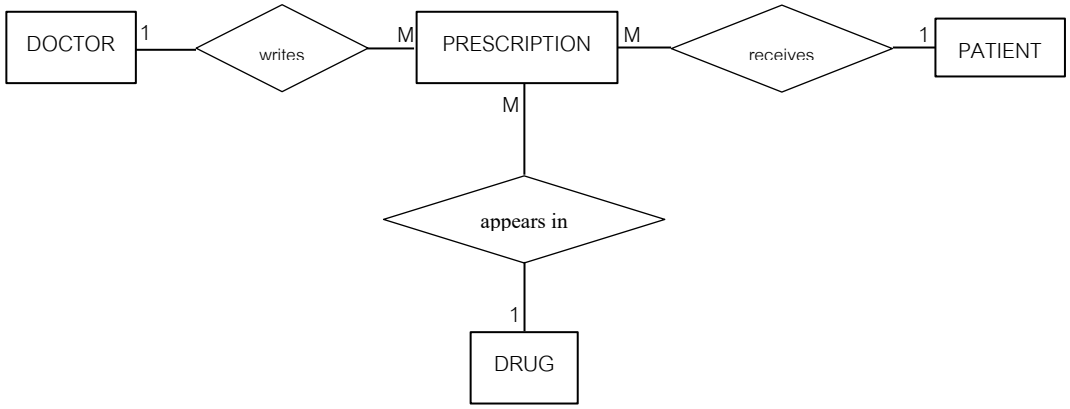
ภาพที่ 5.12 ความสัมพันธ์แบบ Unary
ที่มา : Connolly & Begg (2005, p. 346)

1.2 **ความสัมพันธ์แบบไบนารี (Binary Relationships)** ความสัมพันธ์ชนิดนี้จะมีเอนทิตีที่เกี่ยวข้องกัน 2 เอนทิตี ซึ่งจัดเป็นความสัมพันธ์ชนิดหนึ่งที่พบได้บ่อยในความเป็นจริง ตัวอย่างเช่น คณะ <*ประกอบด้วย*> หลายสาขา ซึ่งแสดงได้ดังภาพที่ 5.13

ภาพที่ 5.13 ความสัมพันธ์แบบ Binary

ที่มา : Connolly & Begg (2005, p. 347)

1.3 **ความสัมพันธ์แบบเทอร์นารี (Ternary Relationships)** เป็นความสัมพันธ์ที่มีเอนทิตีเข้ามาเกี่ยวข้อง 3 เอนทิตี โดยพิจารณาจากภาพที่ 5.14 ดังนี้



ภาพที่ 5.14 ความสัมพันธ์แบบ Ternary

ที่มา : Connolly & Begg (2005, p. 347)

จากภาพที่ 5.14 สามารถอธิบายได้ดังนี้

แพทย์ (DOCTOR) <*เขียน*> ใบสั่งยา (PRESCRIPTION) ได้หลายรายการ

ผู้ป่วย (PATIENT) <*รับ*> ใบสั่งยาได้หลายรายการ

ยา (DRUG) ที่แพทย์ระบุลงไป <*ปรากฏ*> อยู่ในใบสั่งยาได้หลายรายการ

2. โครงสร้างข้อบังคับ

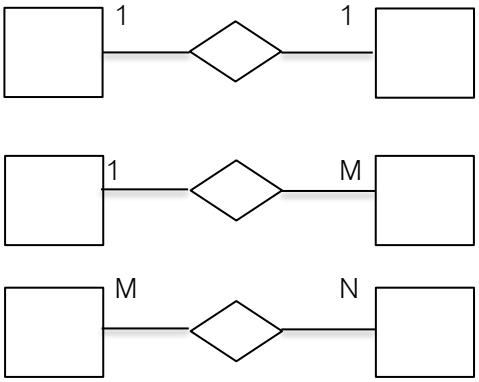
ข้อบังคับในข้อมูลสามารถนำไปใช้เพื่อเป็นส่วนร่วมบนเอนทิตีในความสัมพันธ์ได้ กล่าวคือ ข้อบังคับจะสะท้อนถึงข้อจำกัดต่างๆ บนความสัมพันธ์และใช้เป็นกฎเพื่อไม่ให้เกิดความพยายามฝ่าฝืนข้อบังคับที่สร้างขึ้น

ข้อบังคับ (constraints) เป็นกระบวนการนำความสัมพันธ์และกฎเกณฑ์ทางธุรกิจ (business rules) มาบังคับใช้ เพื่อเป็นเงื่อนไขการจัดเก็บข้อมูลในฐานข้อมูลให้เป็นไปอย่างเหมาะสมและถูกต้อง ตัวอย่างเช่น มหาวิทยาลัยแห่งหนึ่ง ได้มีกฎบังคับไว้ว่าผู้สอนจะต้องสอนหนังสืออย่างน้อย 1 วิชา แต่สอนได้ไม่เกิน 3 วิชา นักศึกษาต้องลงทะเบียนเรียนอย่างน้อย 3 หน่วยกิตต่อภาคการศึกษาและไม่สามารถลงทะเบียนเรียนเกิน 21 หน่วยกิตต่อภาคการศึกษา

ข้อบังคับบนความสัมพันธ์ จะมีอยู่ 2 ชนิดด้วยกันคือ cardinality constraints และ participation constraints (Teorey, 2006, p. 187)

2.1 Cardinality Constraints

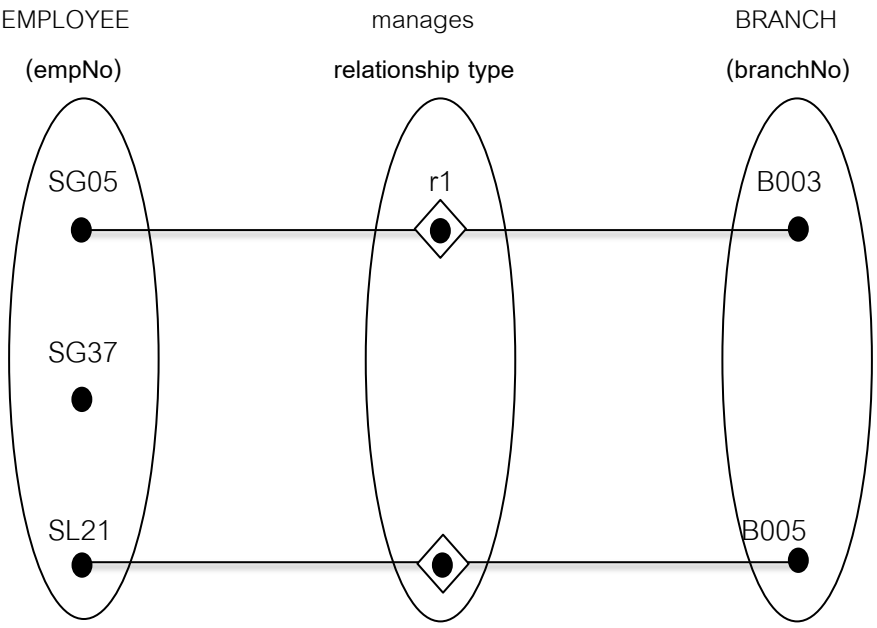
ดีกรีของความสัมพันธ์ส่วนใหญ่มักเป็นในรูปแบบไบนารี และ cardinality ratios ซึ่งอธิบายจำนวนของความสัมพันธ์ที่เป็นไปได้ของแต่ละเอนทิตี สำหรับความสัมพันธ์แบบไบนารีประกอบด้วย one-to-one (1:1), one-to-many (1:M) และ many-to-many (M:N) ซึ่งแสดงได้ดังภาพที่ 5.15



ภาพที่ 5.15 Cardinality Ratios
ที่มา : Teorey (2006, p. 187)

ต่อไปนี้จะนำเสนอความสัมพันธ์ดังกล่าวด้วย Semantic Net Model ซึ่งเป็นแบบจำลองที่แสดงในระดับรายละเอียดของแต่เอนทิตีบนความสัมพันธ์ของข้อมูลได้ชัดเจนยิ่งขึ้น

2.1.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-One Relationships) พิจารณาจากความสัมพันธ์ Manages <*จัดการ*> ที่เกี่ยวข้องกับเอนทิตี EMPLOYEE และ BRANCH จากภาพที่ 5.16 ซึ่งเป็นความสัมพันธ์แบบ One-to-One กล่าวคือ พนักงานคนหนึ่ง <*จัดการ*> หนึ่งสาขา นั่นหมายถึงพนักงานที่เป็นหัวหน้าจะต้อง <*จัดการ*> สาขาหนึ่งสาขา ในขณะที่แต่ละสาขาจะมีหัวหน้าพนักงาน <*จัดการ*> ได้เพียงหนึ่งคน



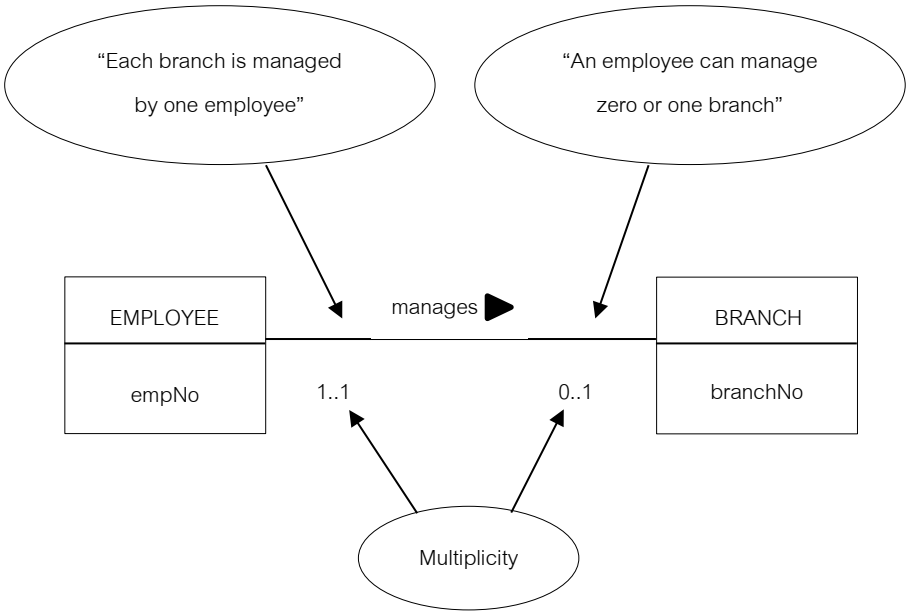
ภาพที่ 5.16 Semantic Net Model กับชนิดความสัมพันธ์แบบ one-to-one
ที่มา : Teorey (2006, p. 187)

จากภาพที่ 5.16 จึงสามารถอธิบายรายระเอียดได้ว่า
พนักงานรหัส SG05 <จัดการ> สาขา B003
พนักงานรหัส SL21 <จัดการ> สาขา B005
พนักงานรหัส SG37 ไม่ได้รับมอบหมายให้ <จัดการ> สาขาใดๆ

หมายความว่า พนักงานคนหนึ่งสามารถ <จัดการ> หนึ่งสาขาหรือไม่ก็ได้ ซึ่งเป็นไปในลักษณะ zero or one (0,1) และแต่ละสาขาจะถูก <จัดการ> โดยพนักงานหนึ่งคน ความสัมพันธ์ดังกล่าวสามารถนำมาเขียนเป็นแผนภาพ E-R ดังภาพที่ 5.17 จากนั้นความสัมพันธ์ดังกล่าวจึงเขียนใหม่ได้ตามแผนภาพ E-R ดังภาพที่ 5.17 ซึ่งสรุปได้ว่า **แต่ละสาขาจะถูกจัดการโดยพนักงานหนึ่งคน และพนักงานจะจัดการสาขาหรือไม่ก็ได้** ซึ่งข้อบังคับดังกล่าวจะทำให้รายละเอียดด้านความสัมพันธ์มีความชัดเจนยิ่งขึ้นแต่อย่างไรก็ตาม ความสัมพันธ์แบบ zero or one นี้คือความสัมพันธ์แบบ 1:1 ตามรูปแบบดั้งเดิมนั่นเอง

ภาพที่ 5.17 ความสัมพันธ์แบบ zero or one
ที่มา : Teorey (2006, p. 188)

นอกจากนี้ ยังสามารถเขียนแบบจำลอง E-R ในรูปแบบของ UML ได้ดังภาพที่ 5.18 อย่างไรก็ตาม ข้อบังคับที่ระบุอยู่ในแผนภาพ UML ที่เป็น 1..1 และ 0..1 จะเรียกว่า **Multiplicity**

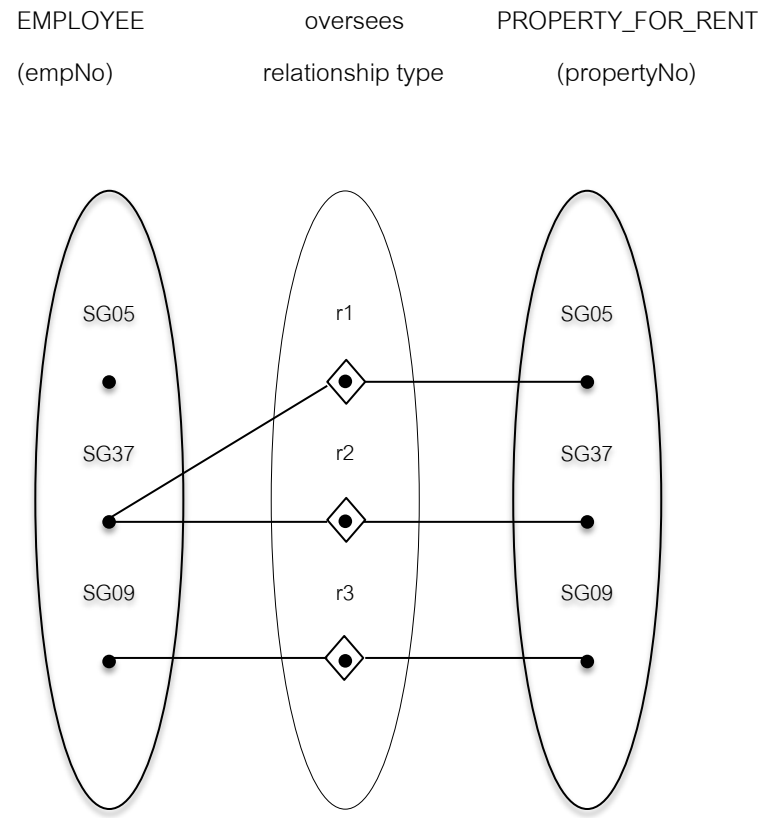


ภาพที่ 5.18 แบบจำลอง E-R แสดงความสัมพันธ์แบบ(1..1),(0..1)ในรูปแบบ UML
ที่มา : Teorey (2006, p. 188)

2.1.2 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One-to-Many Relationship) พิจารณาจากความสัมพันธ์ oversees <ดูแล> ที่เกี่ยวข้องกับเอนทิตี EMPLOYEE และ PROPERTY_FOR_RENT จากภาพที่ 5.19 ซึ่งเป็นความสัมพันธ์แบบ one-to-many โดยพนักงานหนึ่งคน <ดูแล> บ้านเช่าได้มากกว่าหนึ่งหลัง

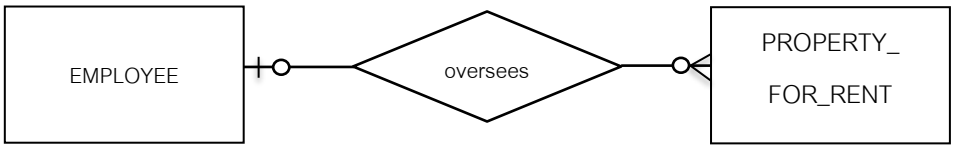
จากภาพที่ 5.19 สามารถอธิบายรายละเอียดได้ดังนี้

1. พนักงานรหัส SG37 <*ดูแล*> บ้านเช่ารหัส PG21 และ PG36
2. พนักงานรหัส SA09 <*ดูแล*> บ้านเช่ารหัส PA14
3. พนักงานรหัส SG05 ไม่ได้ <*ดูแล*> บ้านเช่าใดๆ เลย
4. บ้านเช่ารหัส PG04 ไม่ได้รับการ <*ดูแล*> จากพนักงานคนใดเลย

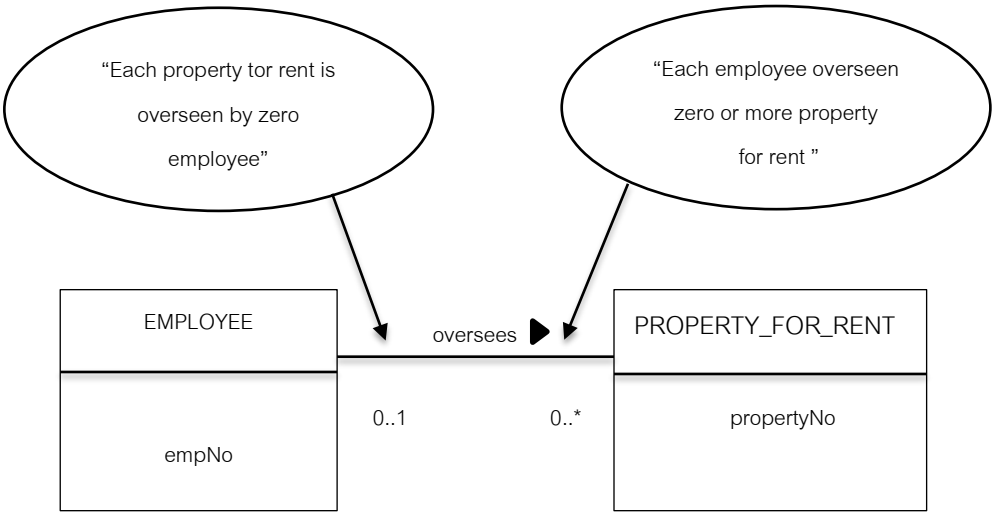


ภาพที่ 5.19 Semantic Net Model กับชนิดความสัมพันธ์แบบ one-to-many
ที่มา : Teorey (2006, p. 189)

นั่นหมายความว่า พนักงานสามารถ <*ดูแล*> บ้านเช่าได้หลายหลังหรือไม่ได้ดูแลก็ได้ ซึ่งเป็นไปในลักษณะ **zero or more** และบ้านเช่าจะถูก <*ดูแล*> โดยพนักงานหนึ่งคนหรือไม่ได้รับการดูแลก็ได้ ซึ่งเป็นไปในลักษณะ **zero or one** ดังนั้นความสัมพันธ์ดังกล่าวจึงเขียนเป็นแบบจำลอง E-R ดังภาพที่ 5.20 ซึ่งสรุปได้ว่า บ้านเช่าแต่ละหลังจะถูกดูแลโดยพนักงานหนึ่งคนหรือไม่ได้รับการดูแลจากพนักงานคนใดเลย และพนักงานแต่ละคนจะดูแลบ้านเช่าได้หลายหลังหรือไม่ได้ดูแลบ้านเช่าเลยก็ได้ โดยอาจเขียนความสัมพันธ์ในรูปแบบ UML ดังภาพที่ 5.21



ภาพที่ 5.20 ความสัมพันธ์ one-to-many ตามข้อบังคับของ zero or one และ zero or more
ที่มา : Teorey (2006, p. 190)



ภาพที่ 5.21 แบบจำลอง E-R แสดงความสัมพันธ์แบบ (0..1),(0..*) ในรูปแบบ UML
ที่มา : Teorey (2006, p. 190)

2.1.3 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many-to-Many Relationship)
พิจารณาจากความสัมพันธ์ advertises <*โฆษณา*> ที่เกี่ยวข้องกับเอนทิตี NEWSPAPER และ PROPERTY_FOR_RENT จากภาพที่ 5.22 ซึ่งเป็นความสัมพันธ์แบบ many-to-many โดยหนังสือพิมพ์หลายฉบับลง <*โฆษณา*> บ้านเช่าได้หลายหลัง

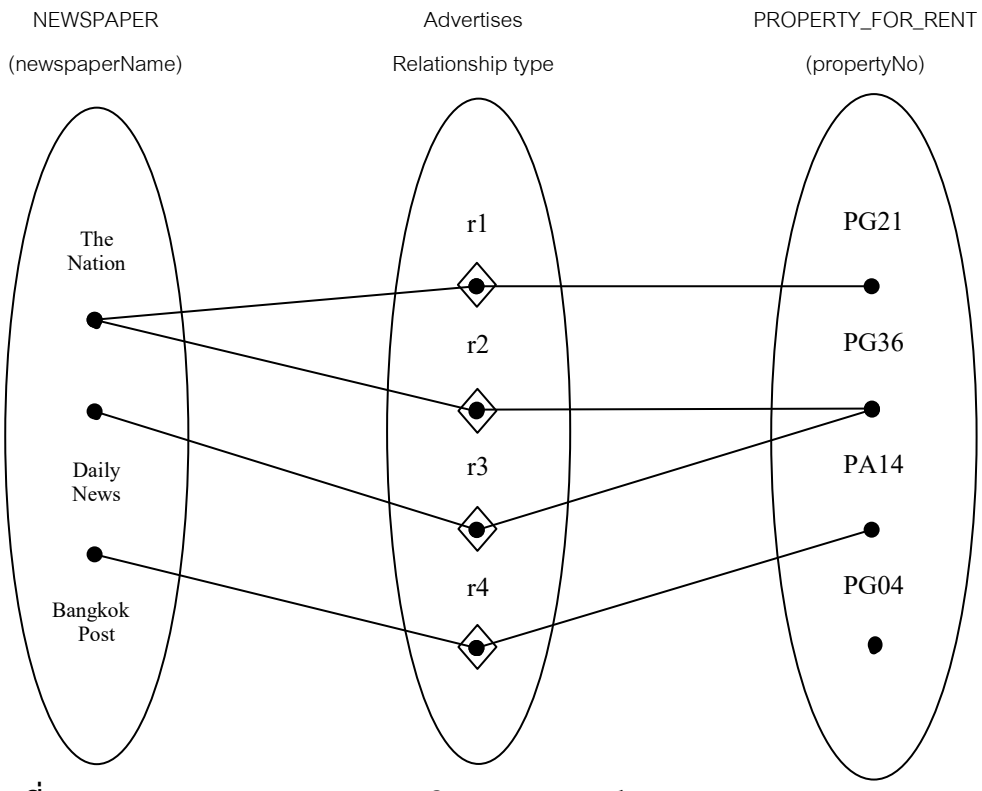
จากภาพที่ 5.22 สามารถอธิบายรายละเอียดได้ดังนี้

หนังสือพิมพ์ The Nation <โฆษณา> บ้านเช่ารหัส PG21 และ PG36

หนังสือพิมพ์ Daily News <โฆษณา> บ้านเช่ารหัส PG36

หนังสือพิมพ์ Bangkok Post <โฆษณา> บ้านเช่ารหัส PA14

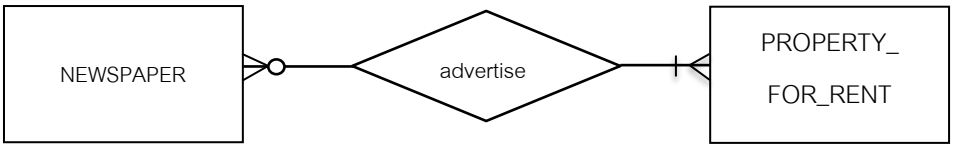
บ้านเช่ารหัส PG04 ไม่ได้ลง <โฆษณา> ในหนังสือพิมพ์ใดๆ เลย



ภาพที่ 5.22 Semantic Net Model กับชนิดความสัมพันธ์แบบ many-to-many

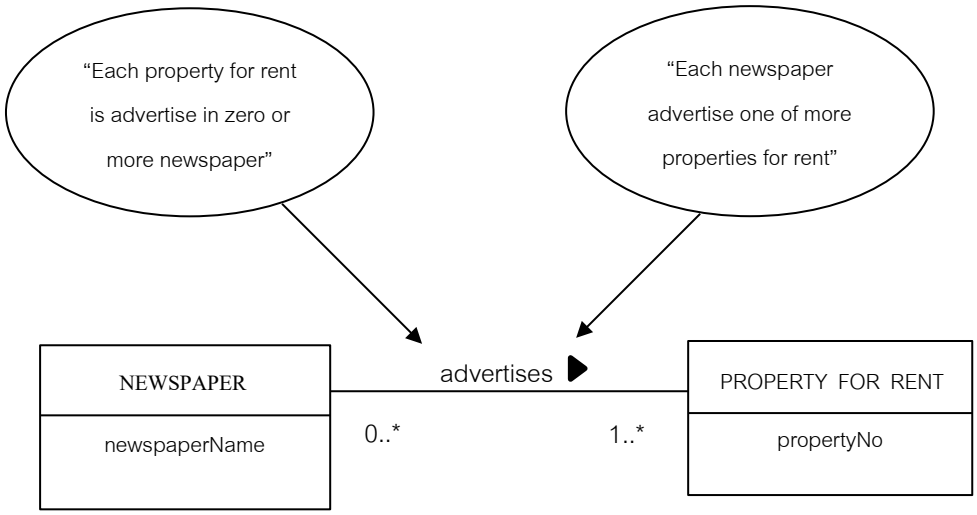
ที่มา : Teorey (2006, p. 191)

หมายความว่า หนังสือพิมพ์แต่ละฉบับ <*โฆษณา*> บ้านเช่าได้หนึ่งหลังหรือมากกว่า ซึ่งเป็นไปในลักษณะ one or more ในขณะที่บ้านเช่าหนึ่งหลังสามารถลง <*โฆษณา*> ในหนังสือพิมพ์ได้หลายฉบับหรือไม่ได้ลง <*โฆษณา*> ในหนังสือพิมพ์เลย ซึ่งเป็นไปในลักษณะ zero or more ดังนั้นความสัมพันธ์ดังกล่าวจึงเขียนตามแผนภาพ E-R ดังภาพที่ 5.23 ซึ่งสรุปได้ว่า บ้านเช่าแต่ละหลังสามารถลงโฆษณาในหนังสือพิมพ์ได้หลายฉบับหรือไม่ได้ลงโฆษณาในหนังสือพิมพ์ใดๆ เลย และหนังสือพิมพ์แต่ละฉบับสามารถลงโฆษณาบ้านเช่าได้หนึ่งหลังหรือมากกว่า ความสัมพันธ์ดังกล่าวในรูปแบบ UML แสดงในภาพที่ 5.24 และสัญลักษณ์ข้อบังคับบนความสัมพันธ์ในรูปแบบต่างๆ แสดงในภาพที่ 5.25



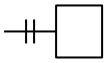
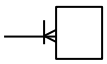
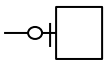
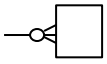
ภาพที่ 5.23 ความสัมพันธ์แบบ many-to-many ตามข้อบังคับของ zero or more และ one or more

ที่มา : Teorey (2006, p. 192)



ภาพที่ 5.24 ความสัมพันธ์แบบ (0..*). (1..*) ในรูปแบบ UML

ที่มา : Teorey (2006, p. 200)

Cardinality Interpretation	Graphic Notation	Multiplicity Constraints (UML)
Mandatory one (one and only one)		1..1
Mandatory Many (one of more)		1..*
zero or one		0..1
zero or more		0..*

ภาพที่ 5.25 สัญลักษณ์ข้อบังคับบนความสัมพันธ์ในรูปแบบต่างๆ (Cardinality Notations)

ที่มา : Teorey (2006, p. 200)

2.2 Participation Constraints

Participation Constraints คือข้อบังคับของการมีส่วนร่วม ซึ่งปกติจะมีอยู่สองชนิดด้วยกันคือ ***Total Participation*** และ ***Partial Participation*** โดยการมีส่วนร่วมแบบ total เป็น

ความสัมพันธ์เชิงบังคับ ในขณะที่การมีส่วนร่วมแบบ Partial นั้นเป็นความสัมพันธ์ในรูปแบบที่เลือกได้

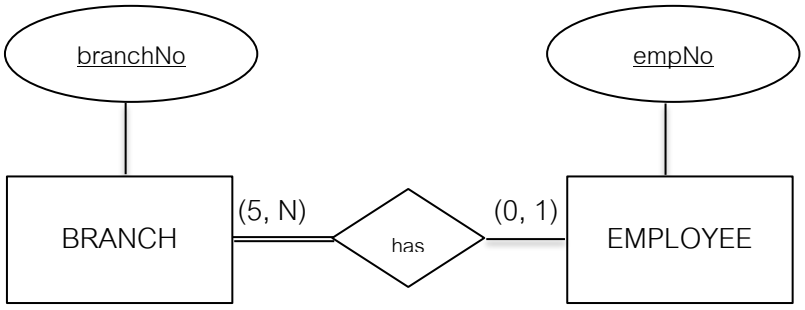
ข้อบังคับของการมีส่วนร่วม (Participation Constraints) หมายถึงการที่เอนทิตีหนึ่งยังคงชีพอยู่ได้ด้วยการพึ่งพิงจากเอนทิตีหนึ่ง ตัวอย่างเช่น เอนทิตี BRANCH <*has*> EMPLOYEE หมายความว่า สาขา <มี> พนักงาน โดยสมมติว่าทุกๆ สาขาจะต้องมีการบรรจุพนักงานจึงสามารถเกิดสาขาขึ้นมาได้ กล่าวคือ สาขาจะเกิดไม่ได้เลยหากไม่มีการบรรจุพนักงาน จึงหมายความว่า สาขาจะคงชีพอยู่ได้ก็ต่อเมื่อสาขานั้นมีพนักงานบรรจุอยู่ ดังนั้นการมีส่วนร่วมของเอนทิตี EMPLOYEE ในความสัมพันธ์ของ <*has*> ก็คือแบบ total นั่นเอง อย่างไรก็ตาม สำหรับเอนทิตี EMPLOYEE พนักงานบางคนอาจไม่จำเป็นต้องประจำอยู่สาขาก็ได้ ตัวอย่างเช่น พนักงานขายที่จะต้องติดต่อลูกค้าภายนอกเป็นประจำ ดังนั้นการมีส่วนร่วมของเอนทิตี EMPLOYEE ในความสัมพันธ์ของ <*has*> จึงเป็นแบบเลือกได้ นั่นก็คือแบบ partial

ภาพที่ 5.26 เป็นการนำเสนอข้อบังคับของการมีส่วนร่วมในความสัมพันธ์ระหว่างเอ็นติตี้ BRANCH <*has*> EMPLOYEE ซึ่งคำว่า total และ partial ในบางครั้งสามารถนำไปใช้อ้างอิงตรงกับคำว่า ***mandatory*** และ ***optional*** ซึ่งสามารถอธิบายได้ดังนี้

Total Participation หรือ Mandatory คือความสัมพันธ์ชนิดบังคับ

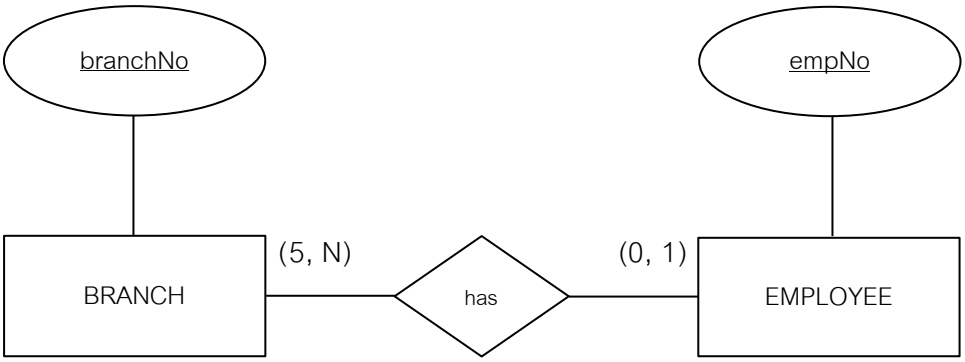
Partial Participation หรือ Optional คือความสัมพันธ์ชนิดเลือกได้

การมีส่วนร่วมในแต่ละความสัมพันธ์จะทำการเชื่อมโยงด้วยเส้น เส้นเดี่ยวหมายถึงการมีส่วนร่วมแบบ partial (optional) ในขณะที่เส้นคู่หมายถึงการมีส่วนร่วมแบบ total (mandatory) ดังตัวอย่างในภาพที่ 5.26 แสดงการมีส่วนร่วมในความสัมพันธ์ของเอนทิตี BRANCH <*has*> EMPLOYEE โดยสาขาจะอยู่ได้ต่อเมื่อมีพนักงาน



ภาพที่ 5.26 การมีส่วนร่วมในความสัมพันธ์ของเอนทิตี BRANCH <*has*> EMPLOYEE
ที่มา : Chittayasothorn (2007, p. 213)

แนวทางอื่นที่สามารถนำมาใช้เป็นข้อบังคับในความสัมพันธ์คือ การแสดงค่าต่ำสุดและค่าสูงสุด (Min, Max) บนเส้นเชื่อมโยงความสัมพันธ์ระหว่างเอนทิตี พิจารณาจากตัวอย่างภาพที่ 5.27 BRANCH <*has*> EMPLOYEE จะเห็นได้ว่าบนเส้นที่ใช้เชื่อมโยงความสัมพันธ์จะมีตัวเลขแสดงค่า (Min, Max) กำกับอยู่ ซึ่งมีความหมายว่า เอนทิตี BRANCH จะต้องมีจำนวนพนักงานอย่างน้อย 5 คน (Min=5) โดยไม่จำกัดจำนวนพนักงานสูงสุด (Max=N) และในทำนองเดียวกันค่าตัวเลข (0,1) ของเอนทิตี EMPLOYEE หมายความว่าพนักงานอาจไม่ได้ประจำอยู่ในสาขา (Min=0) กับพนักงานที่ต้องประจำอยู่ในสาขาที่ตนสังกัดอยู่ (Max=1) (Chittayasothorn, 2007, p. 213)



ภาพที่ 5.27 ข้อบังคับการมีส่วนร่วมในความสัมพันธ์ของเอนทิตี BRANCH <*has*> EMPLOYEE

ที่มา : Chittayasothorn (2007, p. 213)

3. ปัญหาที่เกิดขึ้นในแบบจำลอง E-R

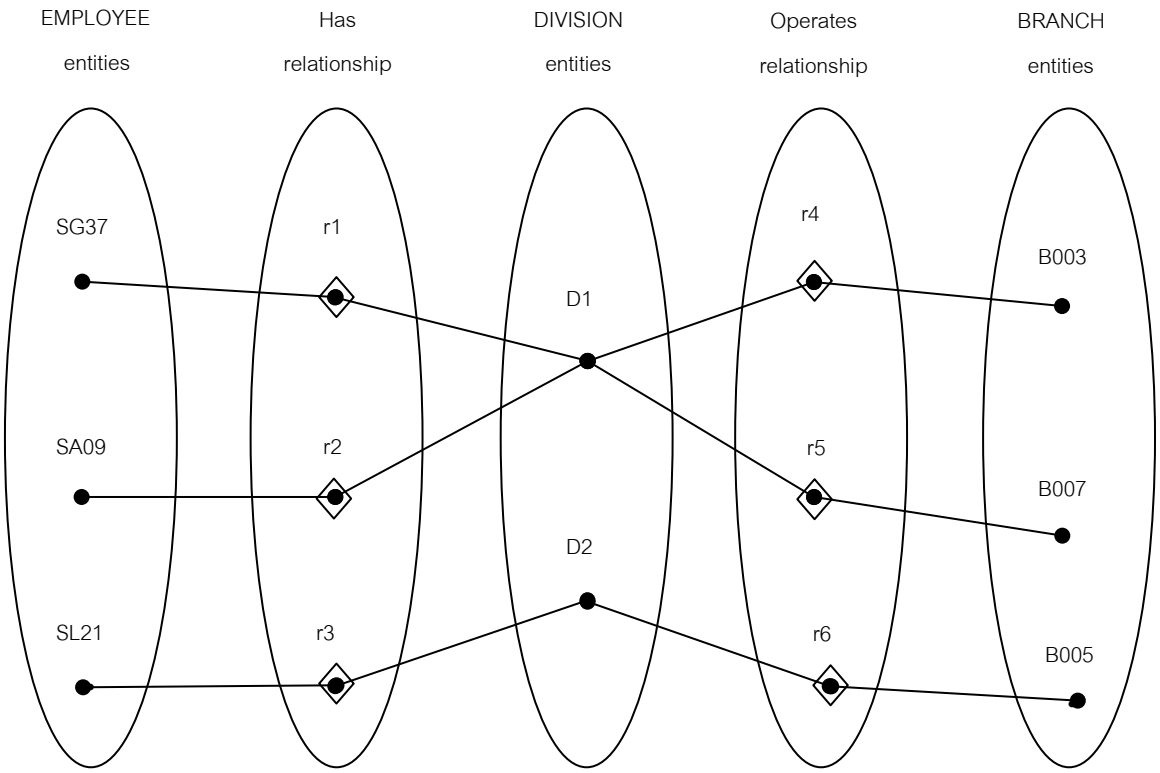
การเขียนแบบจำลอง E-R นั้น จัดเป็นขั้นตอนการออกแบบในระดับตรรกะ ดังนั้นจึงอาจเกิดปัญหาบางอย่างได้ในระยะของการออกแบบ โดยเฉพาะหลุมพรางในการสร้างความสัมพันธ์ระหว่างเอนทิตี ซึ่งก่อให้เกิดการแปลความหมายความสัมพันธ์ผิดพลาด โดยความผิดพลาดหรือปัญหาในแบบจำลอง E-R จะมีอยู่สองรูปแบบด้วยกันคือ Fan Traps และ Chasm Traps (Coronel, Morris & Rob, 2013, p. 224)

3.1 Fan Traps

Fan Traps เป็นปัญหาที่ก่อให้เกิดความกำกวมหรือไม่ชัดเจนในการแสดงข้อมูลที่เราสนใจ ซึ่งปัญหาดังกล่าวเป็นปัญหาที่เกี่ยวข้องกับการจัดความสัมพันธ์ของเอนทิตีที่มักพบความสัมพันธ์แบบ one-to-many หากปัญหาที่ซ่อนเร้นดังกล่าวสามารถค้นพบได้ในระยะของการออกแบบแบบจำลอง โดยหากพบปัญหา ก็เพียงแต่ปรับโครงสร้างแบบจำลองเสียใหม่เพื่อขจัดปัญหาดังกล่าวและทำให้ถูกต้อง แต่หากพบปัญหานี้ในระยะที่มีการนำไปใช้แล้วก็อาจจะส่งผลกระทบต่อการเรียกดูข้อมูลได้

ภาพที่ 5.28 ตัวอย่างปัญหาของ Fan Traps ในแบบจำลอง E-R

ที่มา : Coronel, Morris & Rob (2013, p. 224)



ภาพที่ 5.29 รายละเอียดปัญหา Fan Traps ในแบบจำลอง E-R ที่นำเสนอด้วย Semantic Net Model

ที่มา : Coronel, Morris & Rob (2013, p. 224)

จากภาพที่ 5.28 จะเห็นว่ามีฝ่ายงานหนึ่ง (DIVISION) ปฏิบัติการ <*operates*> อยู่ในหลายสาขา (BRANCH) ขณะเดียวกันฝ่ายงานนั้นก็มีพนักงานสังกัดอยู่หลายคน

จาก Semantic Net Model ดังภาพที่ 5.29 จะพบปัญหาที่ซ่อนเร้นอยู่ก็คือแล้วจะทราบได้อย่างไรว่าพนักงานคนนี้ทำงานอยู่ที่สาขาใด

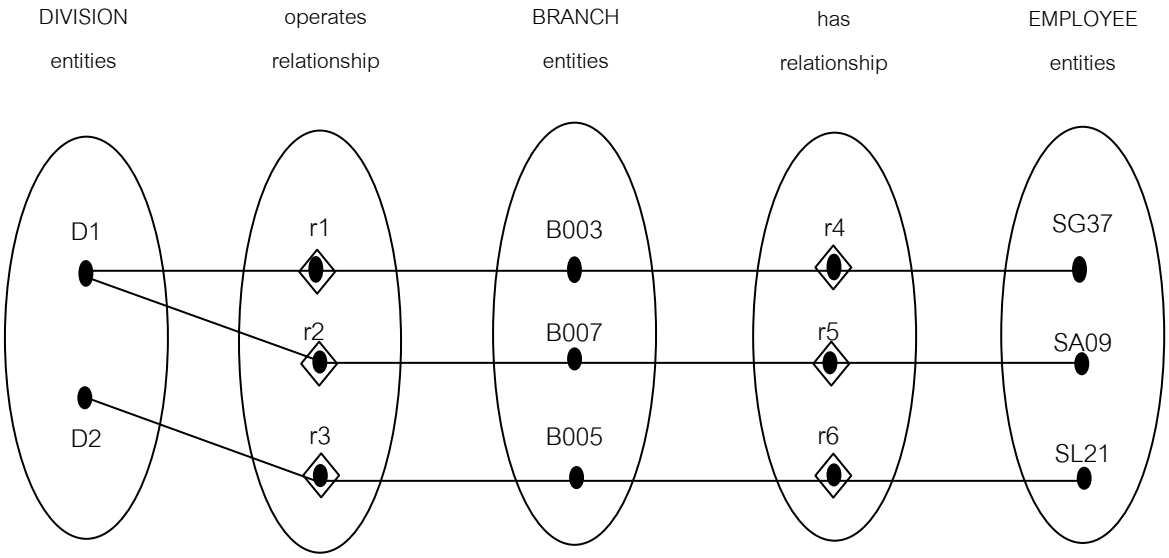
คำถาม พนักงานรหัส SG37 ทำงานอยู่สาขาใด

คำตอบ B003 หรือ B007

จากคำถามดังกล่าว เราไม่สามารถได้รับคำตอบที่แน่ชัดได้เลย ดังนั้นข้อผิดพลาดนี้จึงต้องได้รับการแก้ไขด้วยการปรับโครงสร้าง ด้วยการจัดเรียงความสัมพันธ์ของเอนทิตีเสียใหม่ ดังภาพที่ 5.30

ภาพที่ 5.30 แบบจำลอง E-R ที่ปรับโครงสร้างใหม่ เพื่อขจัดปัญหา Fan Traps

ที่มา : Coronel, Morris & Rob (2013, p. 225)



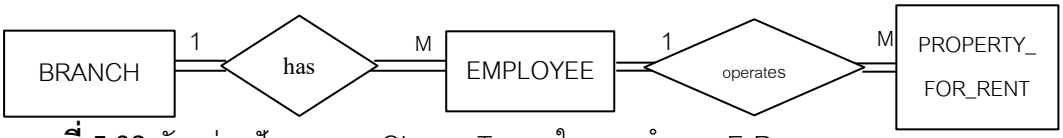
ภาพที่ 5.31 ผลลัพธ์ที่ได้จากการปรับโครงสร้างและขจัดปัญหา Fan Traps

ที่มา : Coronel, Morris & Rob (2013, p. 226)

รายละเอียดข้อมูลใน Semantic Net Model ที่ได้ปรับโครงสร้างใหม่และขจัดปัญหา Fan Traps ออกไปแล้ว ทำให้ทราบว่าพนักงานแต่ละคนทำงานอยู่สาขาใด และปฏิบัติงานที่ฝ่ายงานใด ดังภาพที่ 5.31 จะสามารถให้คำตอบได้แล้วว่า พนักงานรหัส SG37 ทำงานอยู่สาขา B003 โดยปฏิบัติงานอยู่ในฝ่ายงาน D1 นั่นเอง

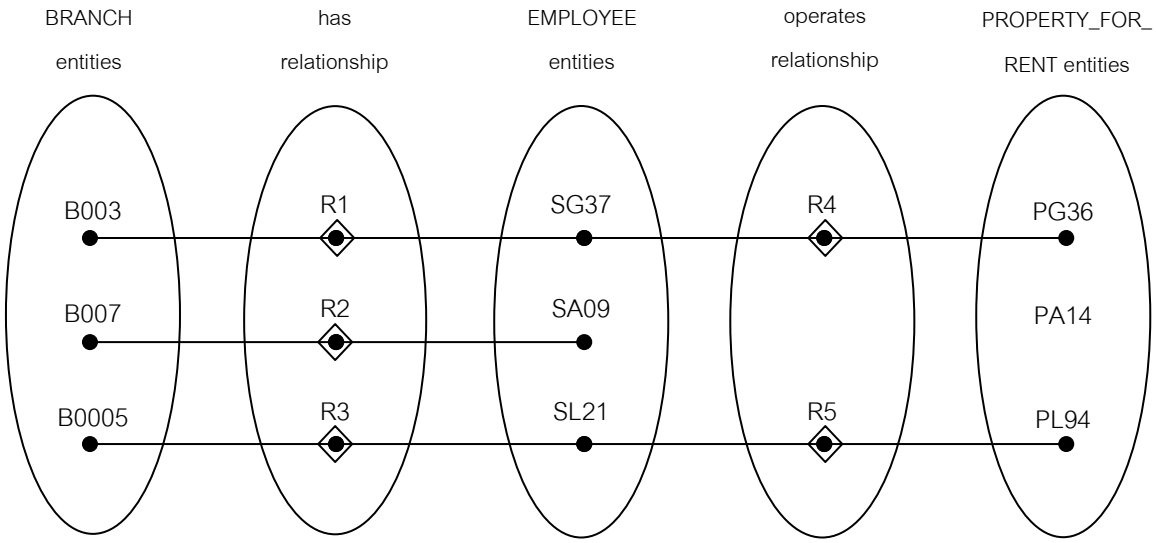
3.2 Chasm Traps

Chasm Traps เป็นปัญหาที่เกิดขึ้นจากการไม่ได้โยงความสัมพันธ์ระหว่างเอนทิตี จึงส่งผลให้ความสัมพันธ์นั้นขาดหายไป ซึ่งปัญหาดังกล่าวมักจะปรากฏบนความสัมพันธ์แบบ partial ในรูปแบบของเส้นความสัมพันธ์นั้นไม่ได้เชื่อมโยงกับเอนทิตีที่เกี่ยวข้องตามที่ควรจะเป็น พิจารณาจากภาพที่ 5.32 ที่อธิบายได้ว่า สาขาหนึ่ง <*มี*> พนักงานได้หลายคน โดยพนักงานแต่ละคนจะ <*ดูแล*> บ้านเช่าที่ตนได้รับมอบหมาย และจากการมีส่วนรวมในความสัมพันธ์ จะทราบได้ว่า ไม่ใช่พนักงานทุกคนต้องดูแลบ้านเช่า และไม่ใช่บ้านเช่าทุกหลังจะต้องถูกดูแลโดยพนักงาน



ภาพที่ 5.32 ตัวอย่างปัญหาของ Chasm Traps ในแบบจำลอง E-R

ที่มา : Coronel, Morris & Rob (2013, p. 227)



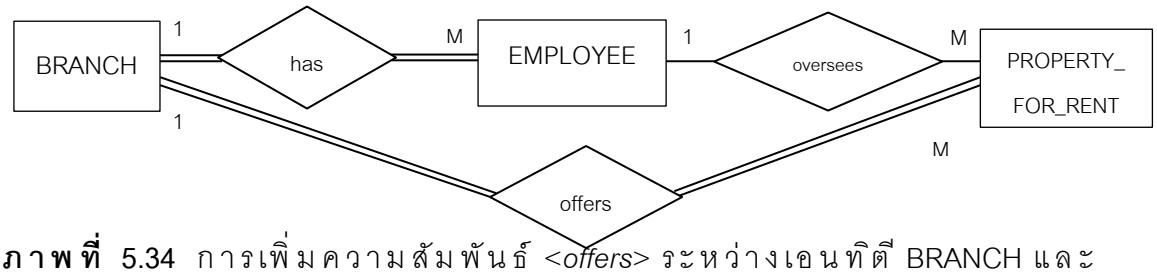
ภาพที่ 5.33 รายละเอียดปัญหา Chasm Traps ที่นำเสนอด้วย Semantic Net Model

ที่มา : Coronel, Morris & Rob (2013, p. 227)

จากรายละเอียดข้อมูลในภาพที่ 5.33 จะพบปัญหาที่ซ้อนเร้นอยู่คือ แล้วจะทราบได้อย่างไรว่าบ้านเช่าหลังนี้อยู่ในสาขาใด

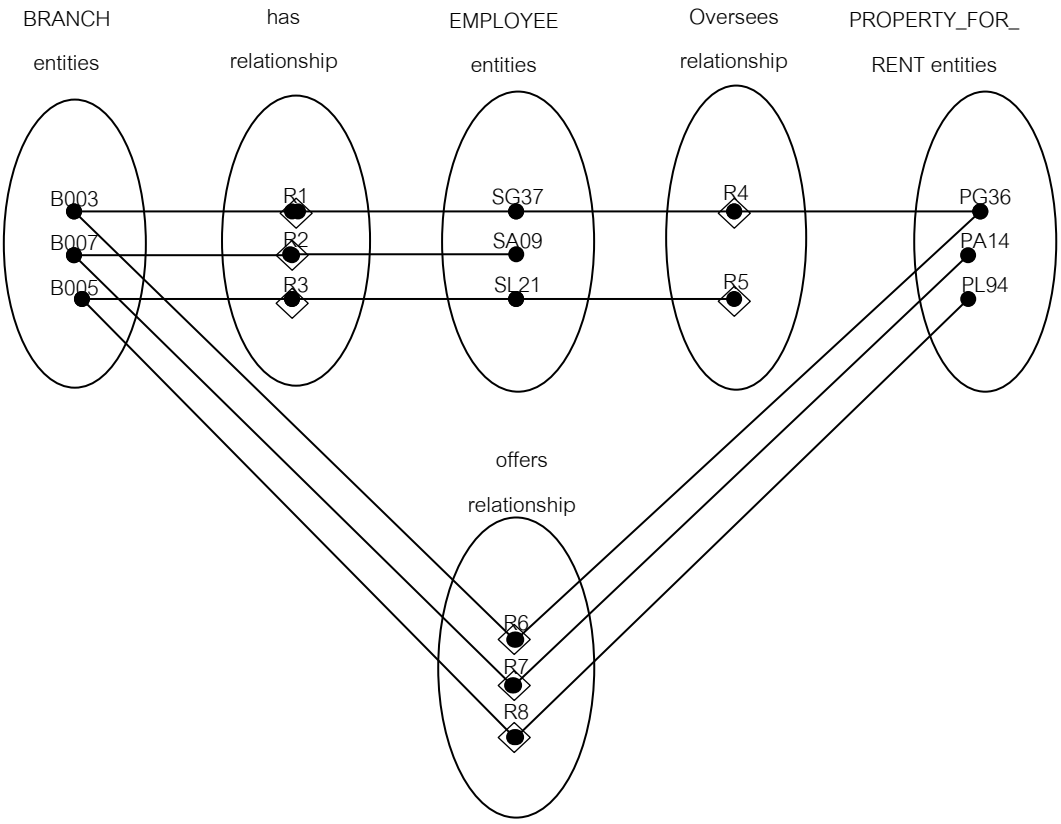
คำถาม บ้านเช่ารหัส PA14 อยู่สาขาใด และพนักงานคนใดเป็นผู้ดูแล

คำตอบ ไม่ทราบ



ภาพที่ 5.34 การเพิ่มความสัมพันธ์ <*offers*> ระหว่างเอนทิตี BRANCH และ PROPERTY_FOR_RENT เพื่อขจัดปัญหา Chasm Traps

ที่มา : Coronel, Morris & Rob (2013, p. 228)



ภาพที่ 5.35 การเชื่อมโยงความสัมพันธ์ระหว่างเอนทิตี

ที่มา : Coronel, Morris & Rob (2013, p. 228)

จากภาพที่ 5.35 หลังจากเชื่อมความสัมพันธ์ <offers> ระหว่างเอนทิตี BRANCH และ PROPERTY_FOR_RENT จึงทำให้ทราบว่าบ้านเช่าแต่ละหลังสังกัดอยู่สาขาใด และมีพนักงานใดเป็นผู้ดูแล

ในความเป็นจริง บ้านเช่าจะเกี่ยวข้องกับสาขาที่สังกัดด้วย และนี่ก็คือผลลัพธ์ของปัญหา Chasm Traps โดยการมีส่วนร่วมแบบ partial ของเอนทิตี EMPLOYEE และ PROPERTY_FOR_RENT ในความสัมพันธ์ <*oversees*> หมายความว่า บ้านเช่าบางหลังไม่สามารถเชื่อมโยงกับสาขาผ่านไปยังพนักงานได้ ดังนั้นการแก้ปัญหานี้ก็คือ จะต้องเชื่อมโยงความสัมพันธ์ <*offers*> ระหว่างเอนทิตี BRANCH และ PROPERTY_FOR_RENT โดยโครงสร้างที่ปรับเปลี่ยนเป็นไปดังภาพที่ 5.34

จากภาพที่ 5.35 ทำให้ทราบว่าบ้านเช่ารหัส PA14 สังกัดสาขา B007 โดยมีพนักงานรหัส SA09 เป็นผู้ดูแลบ้านเช่าหลังนี้

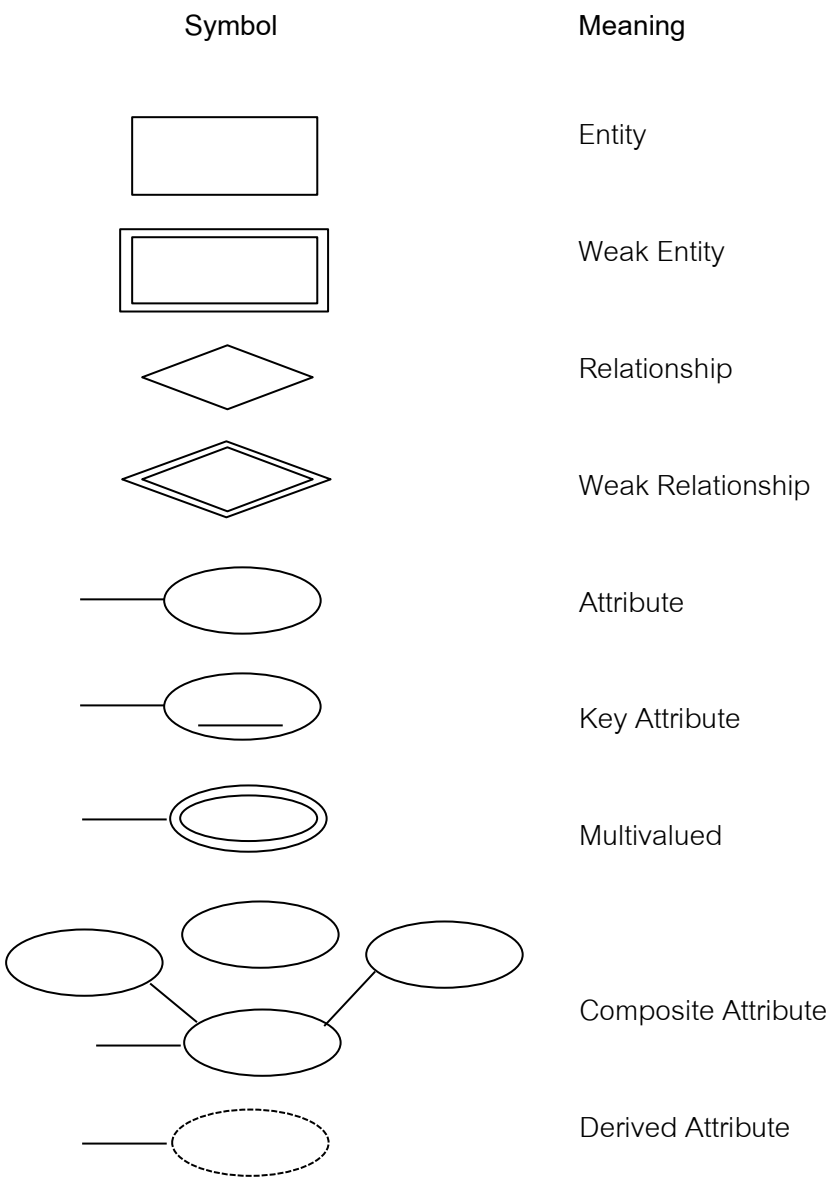
การพัฒนาแบบจำลอง E-R

กระบวนการออกแบบฐานข้อมูลจะเป็นไปในรูปแบบทวนซ้ำมากกว่าที่จะเป็นลำดับตามขั้นตอนแบบเส้นตรง (linear) โดยคำว่า “การทวนซ้ำ (iteration)” หมายความว่า การย้อนกลับมาทำขั้นตอนหรือกิจกรรมก่อนหน้านั้นได้ และหนึ่งในกระบวนการของการทวนซ้ำที่จะนำมาประยุกต์ใช้กับการเขียนแบบจำลอง E-R ปกติมักจะประกอบไปด้วยกิจกรรมต่อไปนี้ (Eric & Jim, 2012, p. 175)

1. ดำเนินการเขียนรายละเอียดการปฏิบัติงานขององค์กรเป็นข้อความบรรยาย
2. ระบุรายละเอียดกฎเกณฑ์ทางธุรกิจว่ามีอะไรบ้าง
3. กำหนดเอนทิตีและความสัมพันธ์จากกฎเกณฑ์ทางธุรกิจ
4. เขียนแบบจำลอง E-R ฉบับเริ่มต้น
5. กำหนดแอตทริบิวต์ที่เพียงพอต่อการอธิบายรายละเอียดของเอนทิตี รวมถึงกำหนดคีย์หลัก
6. ทบทวน ปรับปรุง และตรวจสอบแบบจำลอง E-R ใหม่

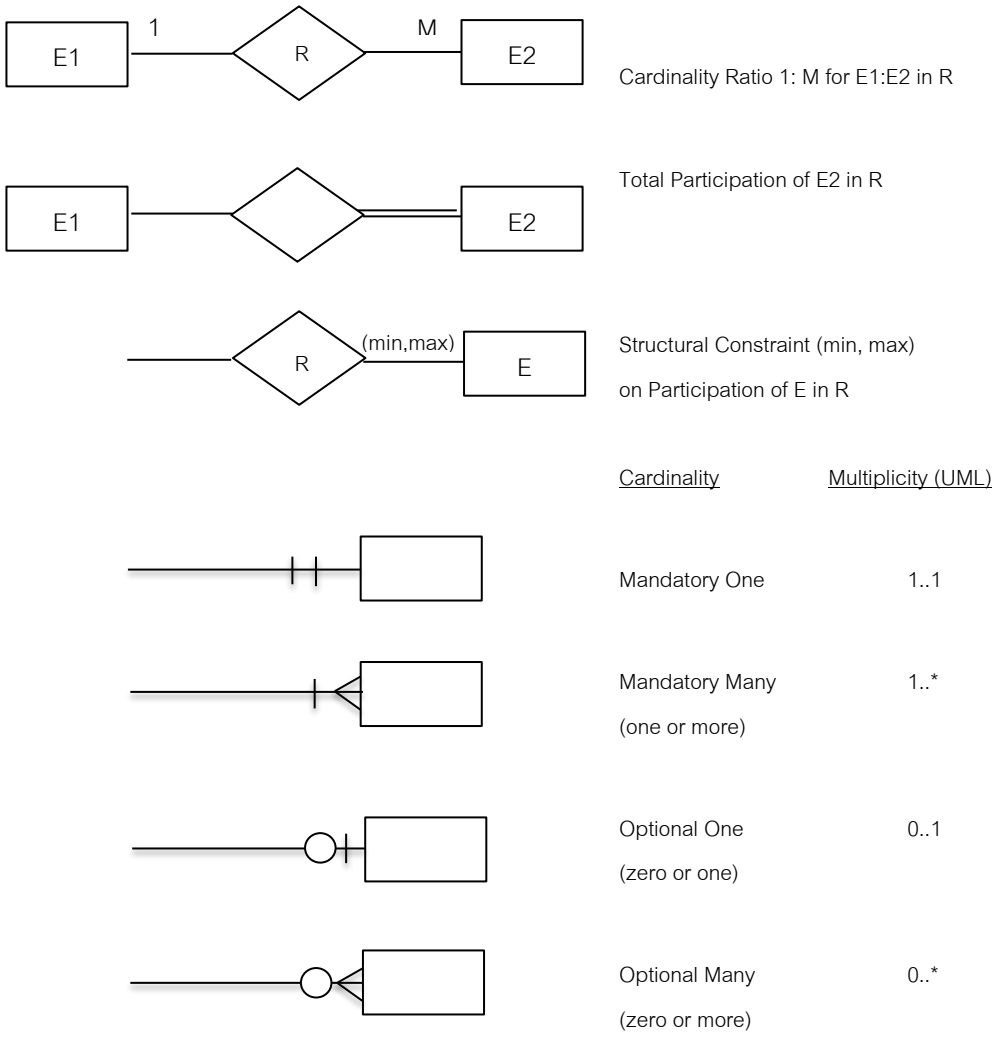
สำหรับกระบวนการตรวจสอบแผนภาพ E-R ในเบื้องต้นอาจพบสิ่งขาดตกบกพร่องไปบ้าง กล่าวคืออาจยังไม่ครอบคลุมความต้องการเท่าที่ควร ดังนั้นกระบวนการตรวจสอบเหล่านี้จึง

เกี่ยวข้องกับการเพิ่มเติมเอนทิตี แอตทริบิวต์ รวมถึงการเชื่อมโยงความสัมพันธ์ ซึ่งการดำเนินการดังกล่าวจะทำให้แบบจำลอง E-R ที่สร้างขึ้นมีความชัดเจนขึ้นเรื่อยๆ โดยกระบวนการตรวจสอบจะทวนซ้ำไปจนกระทั่งผู้ใช้และผู้ออกแบบได้ตกลงพร้อมใจกันว่า แบบจำลอง E-R ที่สร้างขึ้นมานั้นครอบคลุมกิจกรรมและฟังก์ชันในองค์กร



ภาพที่ 5.36 สัญลักษณ์ในแบบจำลอง E-R

ที่มา : Eric & Jim (2012, p. 176)



ภาพที่ 5.36 **(ต่อ)** สัญลักษณ์ในแบบจำลอง E-R
ที่มา : Eric & Jim (2012, p. 176)

ระหว่างกระบวนการออกแบบ นักออกแบบฐานข้อมูลไม่จำเป็นต้องสอบถามหรือสัมภาษณ์ผู้ใช้เพื่อช่วยกำหนดเอนทิตี แอตทริบิวต์ และความสัมพันธ์ทุกครั้งไป เพราะสิ่งที่ช่วยนักออกแบบได้มากก็คือ การรวบรวมตัวอย่างเอกสาร แบบฟอร์มต่างๆ และรายงานที่จัดพิมพ์เป็นประจำภายในหน่วยงาน ข้อมูลเอกสารเหล่านี้ได้มาจากผู้ใช้ตามส่วนงานต่างๆ ซึ่งข้อมูลเหล่านี้จะช่วยนักออกแบบในการเขียนแบบจำลอง E-R ให้มีความสมบูรณ์ยิ่งขึ้น สัญลักษณ์ต่างๆ ที่ใช้ในการเขียนแบบจำลอง E-R แสดงในภาพที่ 5.36

กรณีศึกษาระบบงานธุรกิจเพื่อพัฒนาแบบจำลอง E-R

หลังจากที่ได้ศึกษาเกี่ยวกับแบบจำลอง E-R มาพอสมควรบ้างแล้ว ต่อไปนี้เพื่อให้เข้าใจถึงกระบวนการสร้างแบบจำลอง E-R ยิ่งขึ้น ดังนั้นขอให้พิจารณาตัวอย่างกรณีศึกษาต่อไปนี้

วิทยาลัยระดับอุดมศึกษาแห่งหนึ่งได้ดำเนินการเปิดหลักสูตรระดับปริญญาตรี 2 คณะด้วยกันคือ คณะบริหารธุรกิจและคณะเทคโนโลยีสารสนเทศ คณะบริหารธุรกิจประกอบไปด้วยสาขาวิชาการบัญชี สาขาวิชาการตลาด และสาขาวิชาการจัดการ ส่วนคณะเทคโนโลยีสารสนเทศประกอบด้วยสาขาวิชาเทคโนโลยีสารสนเทศและการสื่อสาร และสาขาวิชาการจัดการระบบสารสนเทศ เนื่องจากวิทยาลัยที่เพิ่งเปิดดำเนินการ ดังนั้นทางผู้บริหารจึงต้องการพัฒนาระบบงานทะเบียนและวัดผลขึ้นมาให้ครอบคลุมระบบงานทะเบียน งานวิชาสอนของอาจารย์ และงานประเมินผล

สำหรับข้อมูลรายละเอียดและกฎเกณฑ์ทางธุรกิจ (business rules) ของวิทยาลัยได้ถูกรวบรวมมาในเบื้องต้น ซึ่งประกอบด้วยรายละเอียดต่อไปนี้

1. หนึ่งคณะวิชาประกอบไปด้วยสาขาวิชาต่างๆ

2. นักศึกษาที่สมัครเรียนเลือกศึกษาหนึ่งคณะและหนึ่งสาขา

3. ค่าหน่วยกิตละค่าธรรมเนียมต่างๆ นักศึกษาจะชำระตามเกณฑ์ของปีที่เข้าศึกษาตลอดจนจบการศึกษา ตัวอย่างเช่น ปีที่เข้าศึกษาคือปี 2551 ดังนั้นค่าหน่วยกิตและค่าธรรมเนียมที่นักศึกษาต้องชำระ จะชำระตามเกณฑ์นี้ตลอด ถึงแม้ว่าปีการศึกษาถัดไปจะขึ้นค่าหน่วยกิตก็ตาม

4. นักศึกษาลงทะเบียนเรียนได้หลายวิชา อย่างน้อย 3 หน่วยกิต แต่ไม่เกิน 21 หน่วยกิต

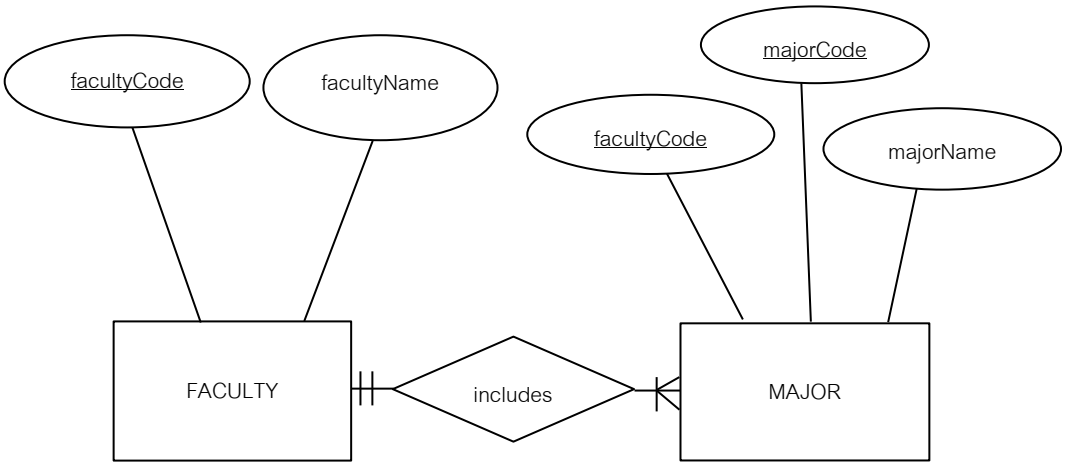
5. บางรายวิชามีเงื่อนไขการลงทะเบียน ซึ่งจะต้องลงทะเบียนเรียนรายวิชาก่อน จึงสามารถลงทะเบียนวิชานั้นได้ เช่น จะต้องเคยเรียนวิชา CS001 มาก่อน จึงสามารถลงทะเบียนวิชา CS102 ได้ เป็นต้น

6. อาจารย์ผู้สอนสามารถสอนได้หลายกลุ่ม (section) หรือหลายรายวิชา โดยอย่างน้อยต้องสอน 1 รายวิชา แต่ไม่เกิน 3 รายวิชา

7. คณะวิชามีหน้าที่ในการว่าจ้างอาจารย์ผู้สอน และอาจารย์ผู้สอนจะสังกัดอยู่ในแต่ละสาขา และต่อไปนี้เป็นแบบจำลอง E-R ที่สร้างขึ้นจากรายละเอียดและกฎเกณฑ์ทางธุระกิจข้างต้น ด้วยการนำเสนอในแต่ละส่วนดังต่อไปนี้

ส่วนที่ 1 : หนึ่งคณะวิชาประกอบไปด้วยสาขาวิชาต่าง ๆ

ทำการกำหนดเอนทิตีคณะ (FACULTY) และสาขา (MAJOR) ขึ้นมา โดยมีความสัมพันธ์แบบ 1:M จากนั้นให้กำหนดแอตทริบิวต์ที่จำเป็น พร้อมระบุคีย์หลัก (PK) ในแต่ละเอนทิตี โดยเอนทิตีคณะ ใช้รหัสคณะ (facultyCode) เป็นคีย์หลัก ส่วนเอนทิตีสาขา ใช้รหัสคณะและรหัสสาขา (facultyCode, majorCode) เป็นคีย์หลัก ดังภาพที่ 5.37

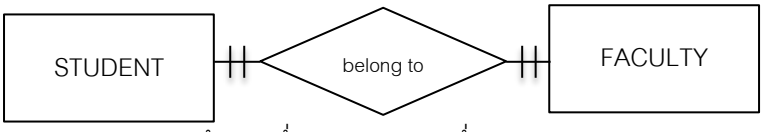


คณะ <*ประกอบด้วย*> หลายสาขา

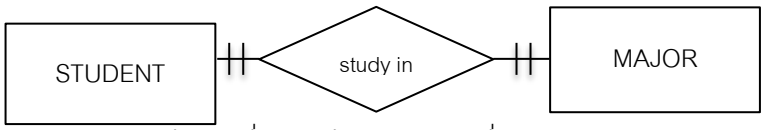
ภาพที่ 5.37 แบบจำลอง E-R ที่แสดงความสัมพันธ์ระหว่างเอนทิตี FACULTY และ MAJOR โดยที่คณะ <*ประกอบด้วย*> หลายสาขา

ส่วนที่ 2 : นักศึกษาที่สมัครเรียนเลือกศึกษาหนึ่งคณะและหนึ่งสาขา

กำหนดเอนทิตีนักศึกษา (STUDENT) ขึ้นมา โดยใช้แอตทริบิวต์รหัสนักศึกษา (stdCode) เป็นคีย์หลัก แล้วทำการเชื่อมโยงกับเอนทิตีคณะและสาขาในรูปแบบความสัมพันธ์แบบ 1:1 ดังภาพที่ 5.38



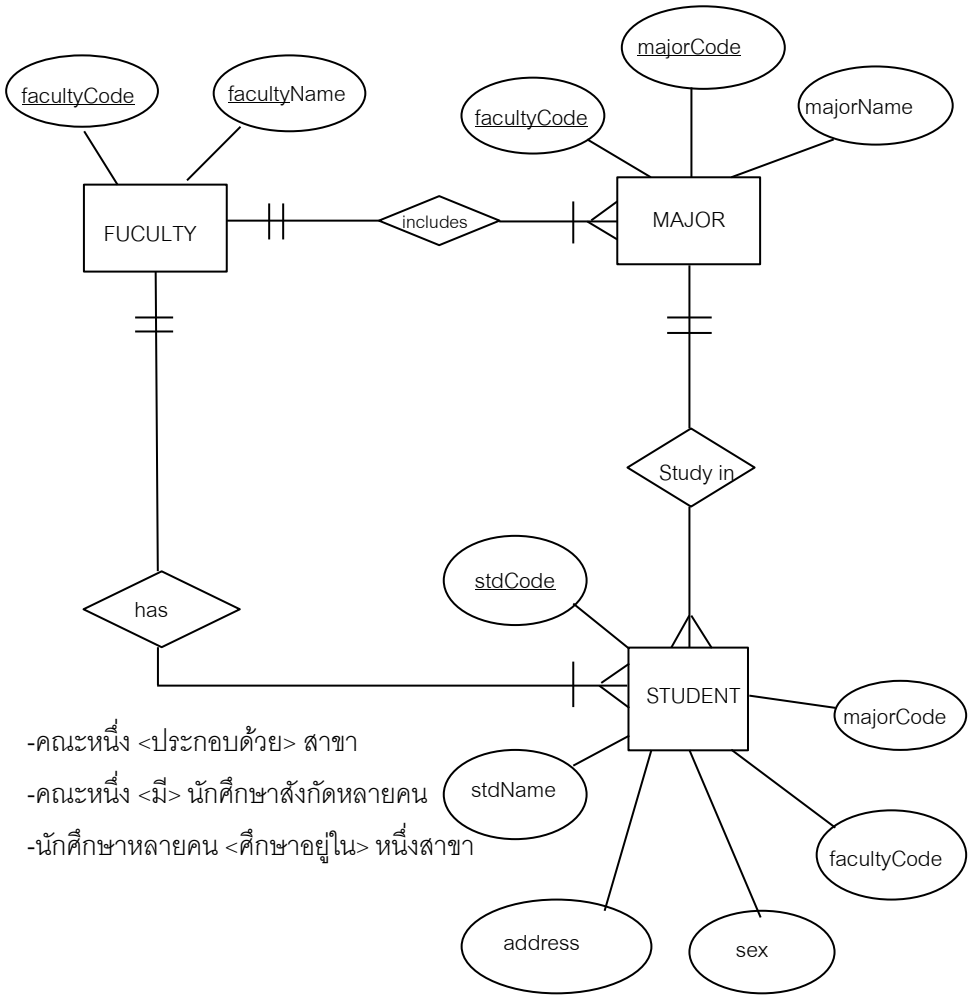
นักศึกษาหนึ่งคน <สังกัด> หนึ่งคณะ



นักศึกษาหนึ่งคน <ศึกษาอยู่ใน> หนึ่งสาขา

ภาพที่ 5.38 แบบจำลอง E-R แสดงความสัมพันธ์แบบ 1:1

อย่างไรก็ตาม ด้วยสัมพันธ์แบบ 1:1 จึงสามารถรวบเอนทิตีทั้งสองเข้าด้วยกัน กล่าวคือ ให้เพิ่มแอตทริบิวต์รหัสคณะ (facultyCode) และรหัสสาขา (majorCode) เข้าในเอนทิตีนักศึกษาได้เลย จากนั้นให้ดำเนินการปรับความสัมพันธ์เสียใหม่เป็นดังภาพที่ 5.39 ดังนี้

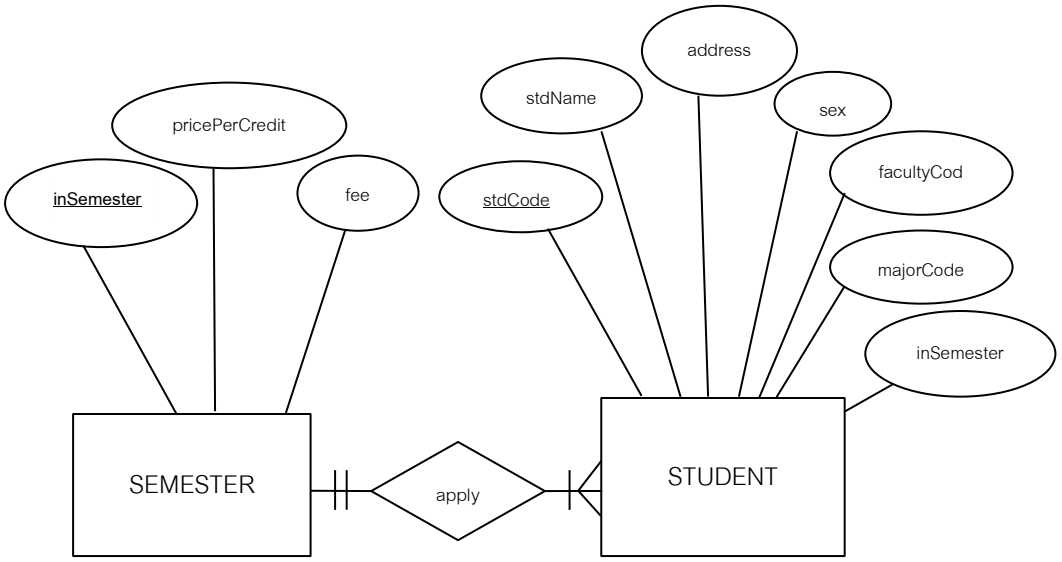


ภาพที่ 5.39 แบบจำลอง E-R แสดงความสัมพันธ์ระหว่างเอนทิตี STUDENT, FACULTY และ MAJOR

ส่วนที่ 3 : คำหน่วยกิตและค่าธรรมเนียมต่าง ๆ นักศึกษาจะชำระตามเกณฑ์ของปีที่เข้าศึกษาตลอดจนจบการศึกษา

กำหนดเอนทิตีภาคการศึกษา (SEMESTR) โดยในที่แอตทริบิวต์ภาคการศึกษา (inSemester) เป็นคีย์หลัก จากนั้นเพิ่มแอตทริบิวต์ inSemester ลงไปในเอนทิตีนักศึกษา เพื่อใช้เป็นคีย์เชื่อมโยงความสัมพันธ์เข้าด้วยกัน

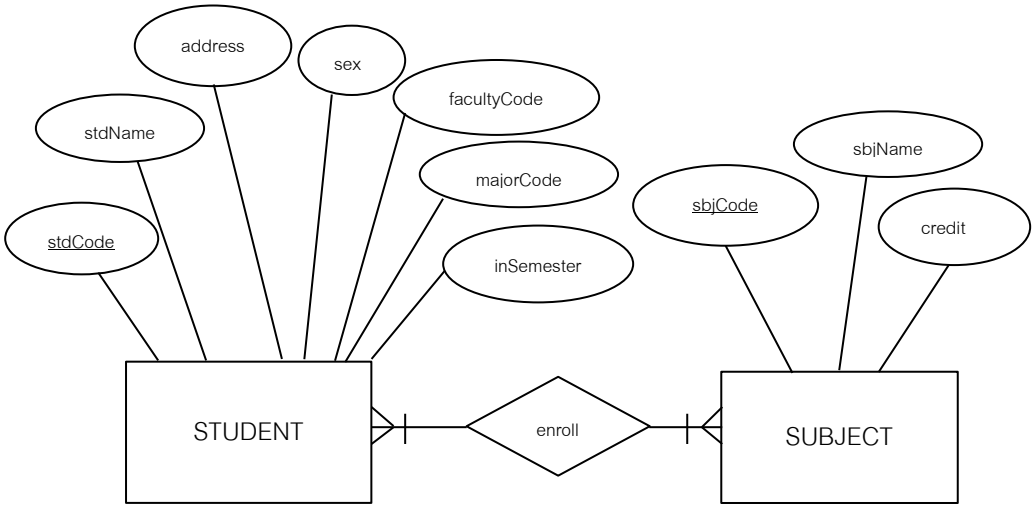
ภาคการศึกษาหนึ่ง เปิดรับสมัครนักศึกษาหลายคน



ภาพที่ 5.40 แบบจำลอง E-R แสดงความสัมพันธ์ระหว่างเอนทิตี SEMESTER และ STUDENT

เอนทิตี SEMESTER จะเก็บรายละเอียดภาคการศึกษาในแต่ละเทอมว่ามีหน่วยกิตและค่าธรรมเนียมต่างๆ เท่าไหร่ ซึ่งปกติจะมีค่าขึ้นในแต่ปีที่แตกต่างกัน ดังนั้นเมื่อเอนทิตี STUDENT มีแอตทริบิวต์ inSemester แล้ว ก็ทำให้ทราบว่านักศึกษารายนั้นได้เข้าศึกษาเมื่อภาคการศึกษาใด และสามารถเชื่อมโยงไปยังเอนทิตี SEMESTER เพื่อตรวจสอบว่าต้องชำระค่าหน่วยกิตหน่วยละเท่าไหร่ ค่าธรรมเนียมเท่าไร ซึ่งจะใช้อัตตราเดิมไปจนกระทั่งจบการศึกษา โดยให้สังเกตเอนทิตี STUDENT ครั้งล่าสุดจะเห็นได้ว่ามีแอตทริบิวต์ที่เป็น FK อยู่ 3 ตัวคือ facultyCode, majorCode และ inSemester ดังภาพที่ 5.40

ส่วนที่ 4 : นักศึกษาจะลงทะเบียนได้หลายวิชา โดยลงทะเบียนได้อย่างน้อย 3 หน่วยกิต แต่ไม่เกิน 21 หน่วยกิต

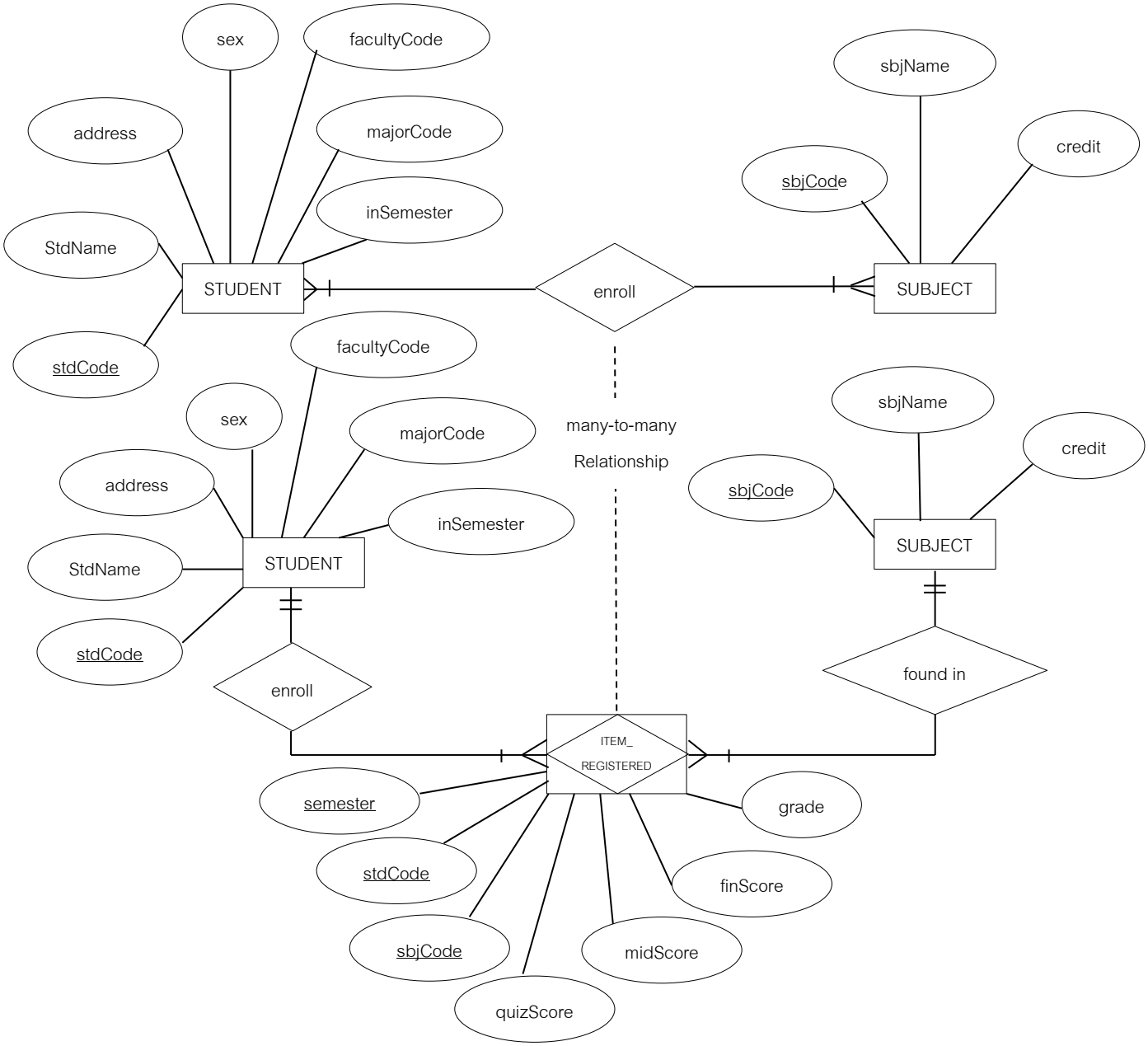


นักศึกษาหลายคน *<ลงทะเบียนเรียน>* หลายวิชา

ภาพที่ 5.41 ความสัมพันธ์แบบ M:N ระหว่างเอนทิตี STUDENT และ SUBJECT

กำหนดเอนทิตีวิชา (SUBJECT) ขึ้นมาและทำการเชื่อมโยงกับเอนทิตี STUDENT ที่มีอยู่ ดังภาพที่ 5.41

อย่างไรก็ตาม โดยธรรมชาติแล้วแบบจำลองเชิงสัมพันธ์จะไม่สนับสนุนความสัมพันธ์แบบ M:N แต่ในขั้นตอนการออกแบบเชิงแนวคิดนั้นสามารถมีขึ้นได้ แต่เมื่อมาถึงกระบวนการออกแบบเชิงตรรกะแล้วควรมีการปรับปรุงใหม่ เนื่องจากความสัมพันธ์แบบ M:N มีการทับซ้อนข้อมูลอยู่ ดังนั้นจึงจำเป็นต้องสร้างเอนทิตีใหม่ขึ้นมา เพื่อแปลงความสัมพันธ์จาก M:N มาเป็น 1:M

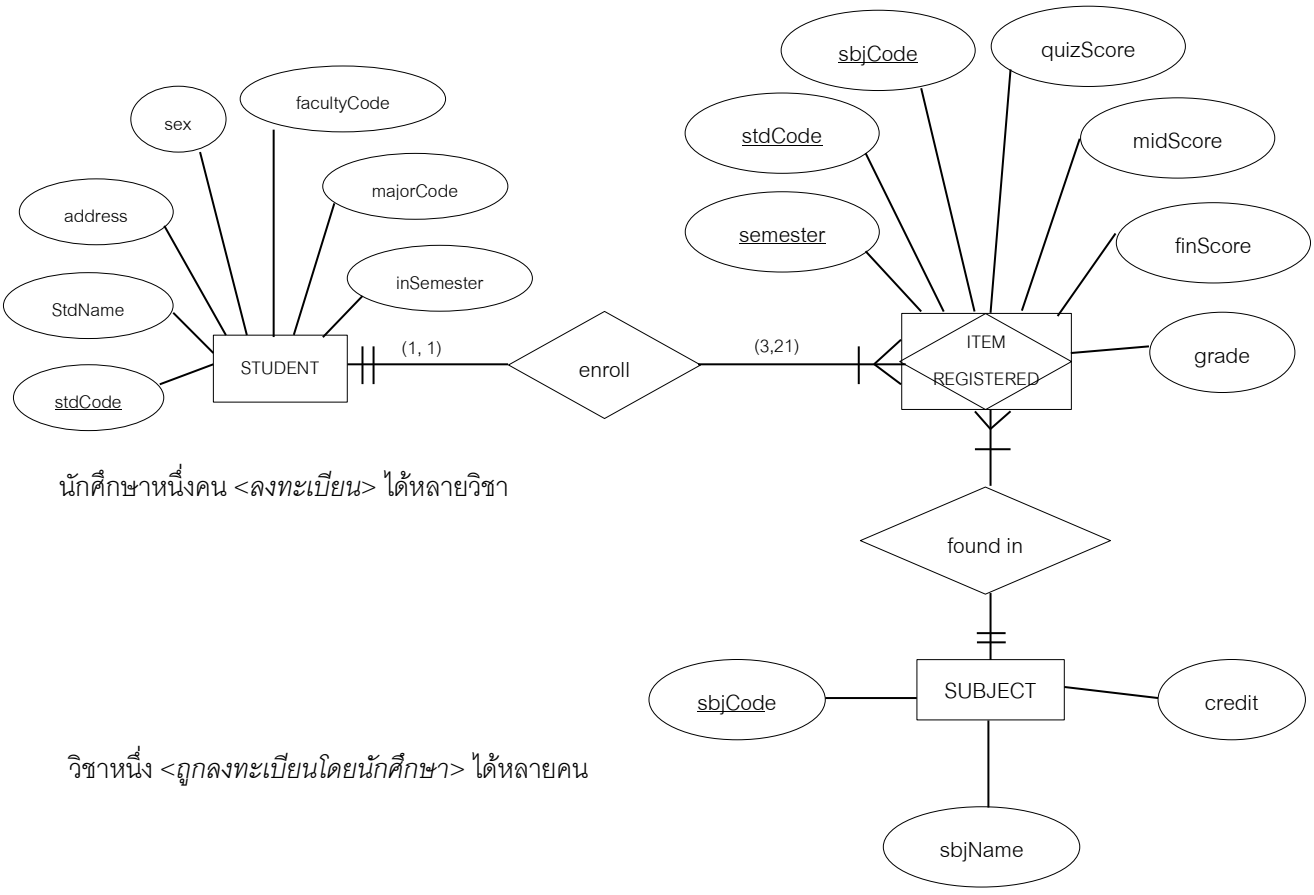


ภาพที่ 5.42 การแก้ปัญหาความสัมพันธ์แบบ M:N ให้เป็นความสัมพันธ์แบบ 1:M

ภาพที่ 5.42 ที่แสดงกระบวนการแก้ปัญหาความสัมพันธ์ M:N มาอยู่ในรูปแบบที่เหมาะสมให้เป็นความสัมพันธ์แบบ 1:M ด้วยการเพิ่มอีกหนึ่งเอนทิตีเข้ามามีส่วนร่วมในความสัมพันธ์ ที่เรียกว่า associative entity โดยเอนทิตีที่สร้างขึ้นมาใหม่ในที่นี้คือ

ITEM_REGISTERED ที่ใช้เก็บข้อมูลนักศึกษาที่ลงทะเบียนเรียนรายวิชาต่างๆ จากนั้นให้กำหนด PK ด้วยการคัดลอก PK ของเอนทิตี STUDENT และ SUBJECT มารวมกัน

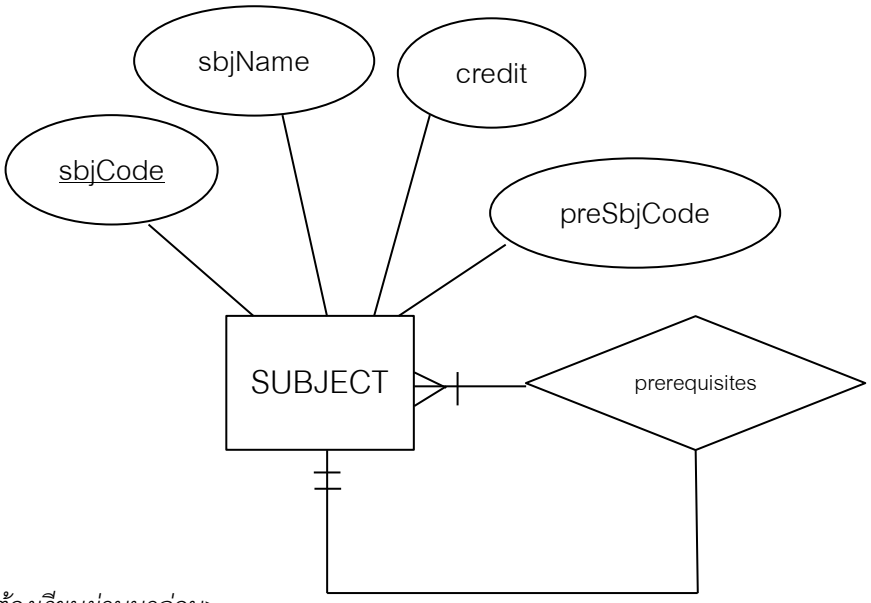
เอนทิตี ITEM_REGISTERED เป็นแหล่งเก็บรายละเอียดวิชาต่างๆ ที่นักศึกษาลงทะเบียน โดยจะมี PK อยู่ 3 แอตทริบิวต์ด้วยกันคือ semester, stdCode และ sbjCode หลายคนอาจสงสัยว่า แอตทริบิวต์ semester เพิ่มขึ้นมาได้อย่างไร ใช้เพียงแค่แอตทริบิวต์ stdCode และ sbjCode ก็เพียงพอแล้วไม่ใช่หรือซึ่งคงต้องย้อนถึงเนื้อหาคุณสมบัติของรีเลชั่นที่ได้เรียนผ่านมา มีกฎอยู่ข้อหนึ่งว่าแต่ละทูเพิลจะต้องไม่มีข้อมูลที่ซ้ำกัน ดังนั้นจึงต้องจำแนกความแตกต่างด้วยคีย์ และเป็นที่เข้าใจว่า นักศึกษาที่ลงทะเบียนเรียนแล้วอาจสอบตกได้ และต้องมาลงทะเบียนเรียนใหม่ ดังนั้นกรณีดังกล่าวหากใช้เพียงแอตทริบิวต์ stdCode และ sbjCode จะทำให้เกิดความซ้ำกันของคีย์ได้ แต่เมื่อมีการเพิ่มแอตทริบิวต์ semester เข้าไป ก็จะทำให้แยกแยะข้อมูลที่แตกต่างกันได้ ดังภาพที่ 5.43



ภาพที่ 5.43 ผลจากการแปลงความสัมพันธ์แบบ M:N มาเป็น 1:M ด้วยการเพิ่มเอนทิตี ITEM_REGISTERED เข้าไป

ส่วนที่ 5 : บางรายวิชามีเงื่อนไขการลงทะเบียน โดยจะจะต้องลงทะเบียนเรียนวิชาก่อน

เงื่อนไขดังกล่าวเป็นในลักษณะรีเคอร์ซีพ ซึ่งเป็นไปตามแบบจำลอง E-R ดังภาพที่ 5.44



ภาพที่ 5.44 ความสัมพันธ์แบบรีเคอร์ซีพของเอนทิตี SUBJECT

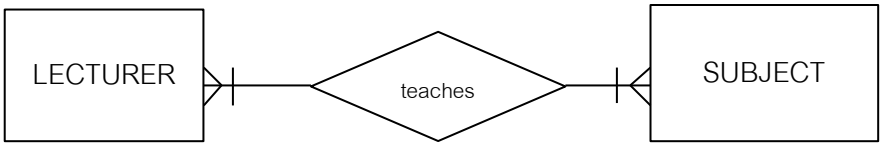
ดังนั้นเอนทิตี SUBJECT จึงต้องเพิ่มแอตทริบิวต์อีกหนึ่งตัวเข้าไป ในที่นี่คือ preSbjCode ซึ่งทำให้เราทราบว่าวิชานี้ต้องเรียนวิชาใดมาก่อน และเพื่อให้เกิดความเข้าใจยิ่งขึ้น ให้พิจารณาจากตัวอย่างข้อมูลในรีเลชั่น SUBJECT ต่อไปนี้

SbjCode	SbjName	Credit	PreSbjCode
CS101	Database Systems	3	
CS103	Systems Analysis and Design	3	CS101

นั่นหมายความว่า หากนักศึกษาคนใดลงทะเบียนเรียนรหัสวิชา CS103 คือวิชา System Analysis and Design จะต้องผ่านการเรียนวิชา Database Systems มาก่อนนั่นเอง

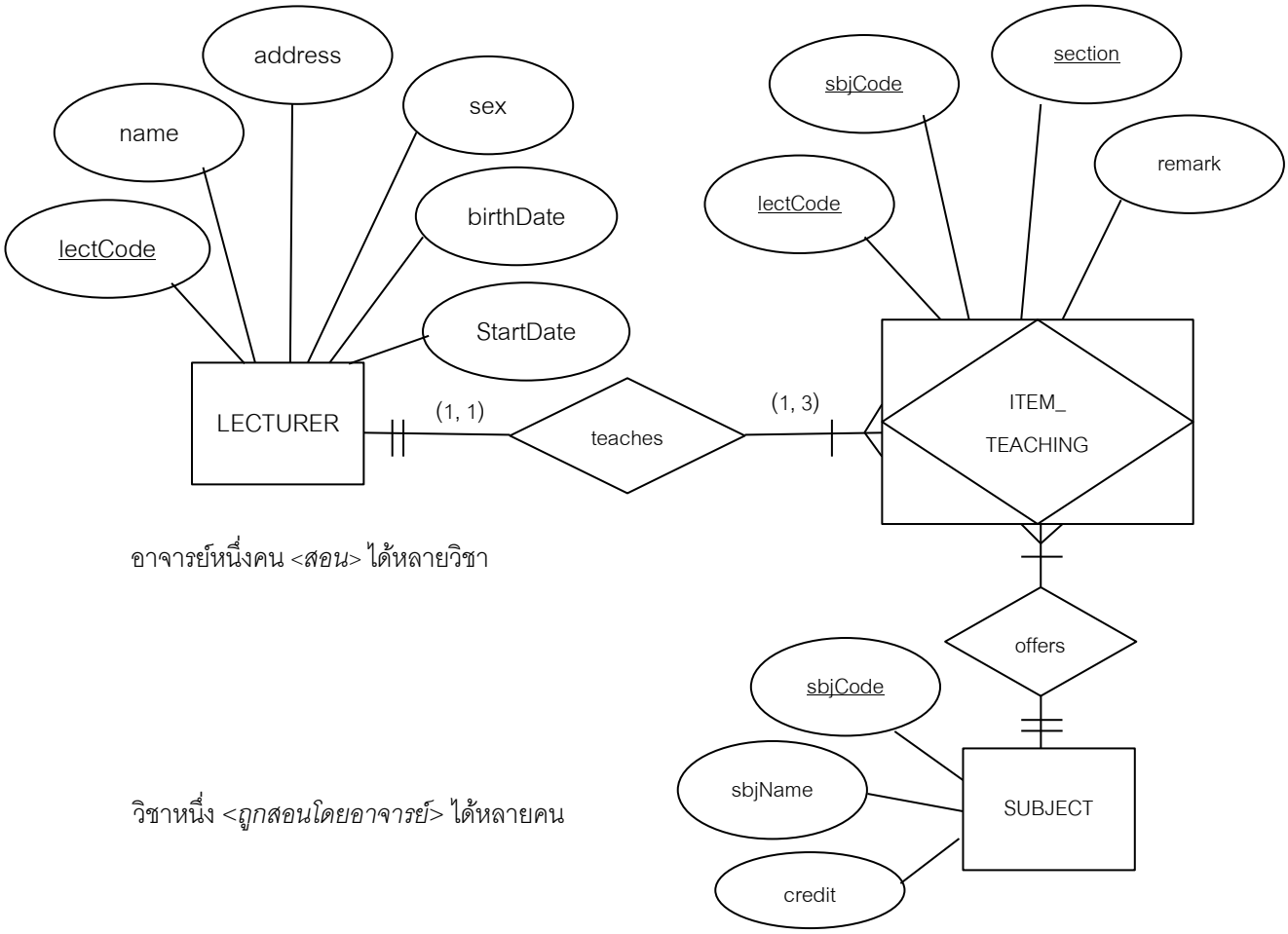
ส่วนที่ 6 : อาจารย์ผู้สอน สามารถสอนได้หลายกลุ่ม (Section) หรือหลายรายวิชา โดยอาจารย์ผู้สอนต้องสอนอย่างน้อย 1 วิชา แต่ไม่เกิน 3 วิชา

กำหนดเอนทิตีอาจารย์ (LECTURER) ขึ้นมา และทำการเชื่อมโยงกับเอนทิตี SUBJECT ที่มีอยู่ ซึ่งเป็นไปได้ดังภาพที่ 5.45



อาจารย์หลายคน <*สอน*> ได้หลายวิชา

ภาพที่ 5.45 แบบจำลอง E-R แสดงความสัมพันธ์แบบ M:N ระหว่างเอนทิตี LECTURER และ SUBJECT



อาจารย์หนึ่งคน <*สอน*> ได้หลายวิชา

วิชาหนึ่ง <*ถูกสอนโดยอาจารย์*> ได้หลายคน

ภาพที่ 5.46 การแปลงความสัมพันธ์แบบ M:N มาเป็น 1:M ด้วยการเพิ่มเอนทิตี ITEM_TEACHING เข้าไป

ในทำนองเดียวกัน เนื่องจากเป็นความสัมพันธ์แบบ M:N ดังนั้น จึงต้องเพิ่มอีกหนึ่งเอนทิตีเข้ามาใหม่ในที่นี้คือเอนทิตี ITEM_TEACHING ที่ใช้จัดเก็บรายวิชาต่างๆ ที่อาจารย์สอน โดยแบบจำลอง E-R ที่ปรับใหม่แสดงไว้ดังภาพที่ 5.46

ส่วนที่ 7 : คณะวิชาว่าจ้างอาจารย์ผู้สอนและอาจารย์ผู้สอนจะสังกัดอยู่ในแต่ละสาขา

การสร้างความสัมพันธ์ให้กับเอนทิตีคณะ (FACULTY) และเอนทิตีอาจารย์ (LECTURER) ซึ่งมีรูปแบบและความสัมพันธ์คือ คณะวิชา <*ว่าจ้าง*> อาจารย์ผู้สอนได้หลายคน จากนั้นให้นำเอนทิตีสาขา (MAJOR) ที่มีอยู่เดิมเข้าไปเชื่อมความสัมพันธ์กับอาจารย์ผู้สอน

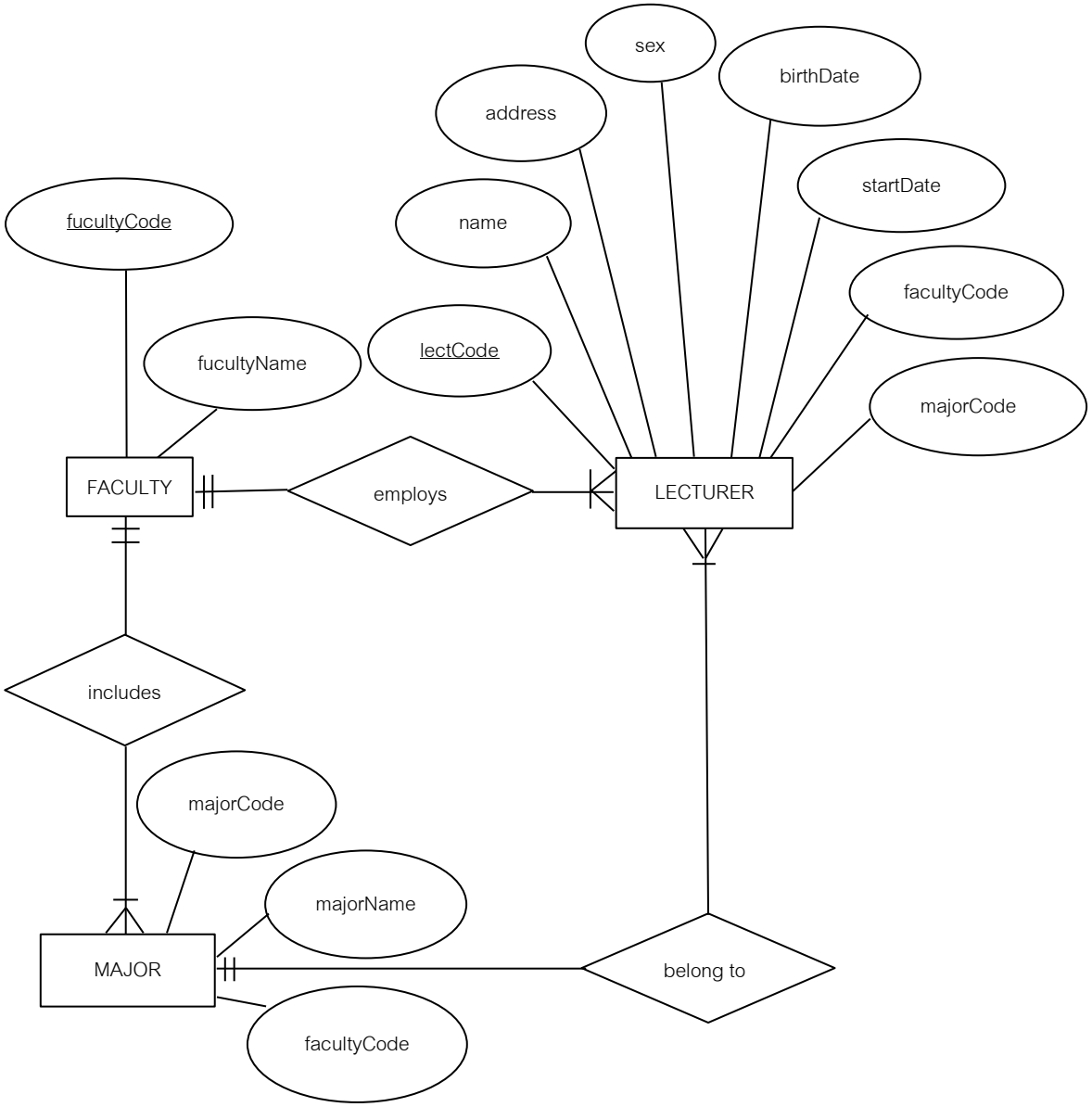
ให้ปรับปรุงคุณสมบัติของเอนทิตี LECTURER ด้วยการเพิ่มแอตทริบิวต์ facultyCode เข้าไป เพื่อเชื่อมโยงกับเอนทิตี FACUTY ในขณะเดียวกันก็เพิ่มแอตทริบิวต์ majorCode เข้าไปด้วย เพื่อทำการเชื่อมโยงความสัมพันธ์ MAJOR ดังภาพที่ 5.47

และจากการออกแบบฐานข้อมูลด้วยการเขียนแบบจำลอง E-R ข้างต้น จึงได้สรุปได้ว่ามีเอนทิตีที่เกี่ยวข้องกับระบบงานดังต่อไปนี้

SEMESTER	ภาคการศึกษา
FACULTY	คณะ
MAJOR	สาขา
STUDENT	นักศึกษา
ITEM_REGISTERED	รายวิชาที่นักศึกษาลงทะเบียน
SUBJECT	วิชา
ITEM_TEACHING	รายวิชาที่อาจารย์สอน
LECTURER	อาจารย์

หลังจากวิชาที่ได้เขียนแบบจำลอง E-R ในแต่ละส่วนเรียบร้อยแล้ว อาจจำเป็นต้องตรวจสอบและดำเนินการปรับปรุงแก้ไขเพิ่มเติมได้จนกระทั่งสมบูรณ์ จากนั้นให้นำแบบจำลอง E-R แต่ละส่วนมาประกอบกันเป็นแบบจำลองรวม (Global Data Model) ดังภาพที่ 5.48

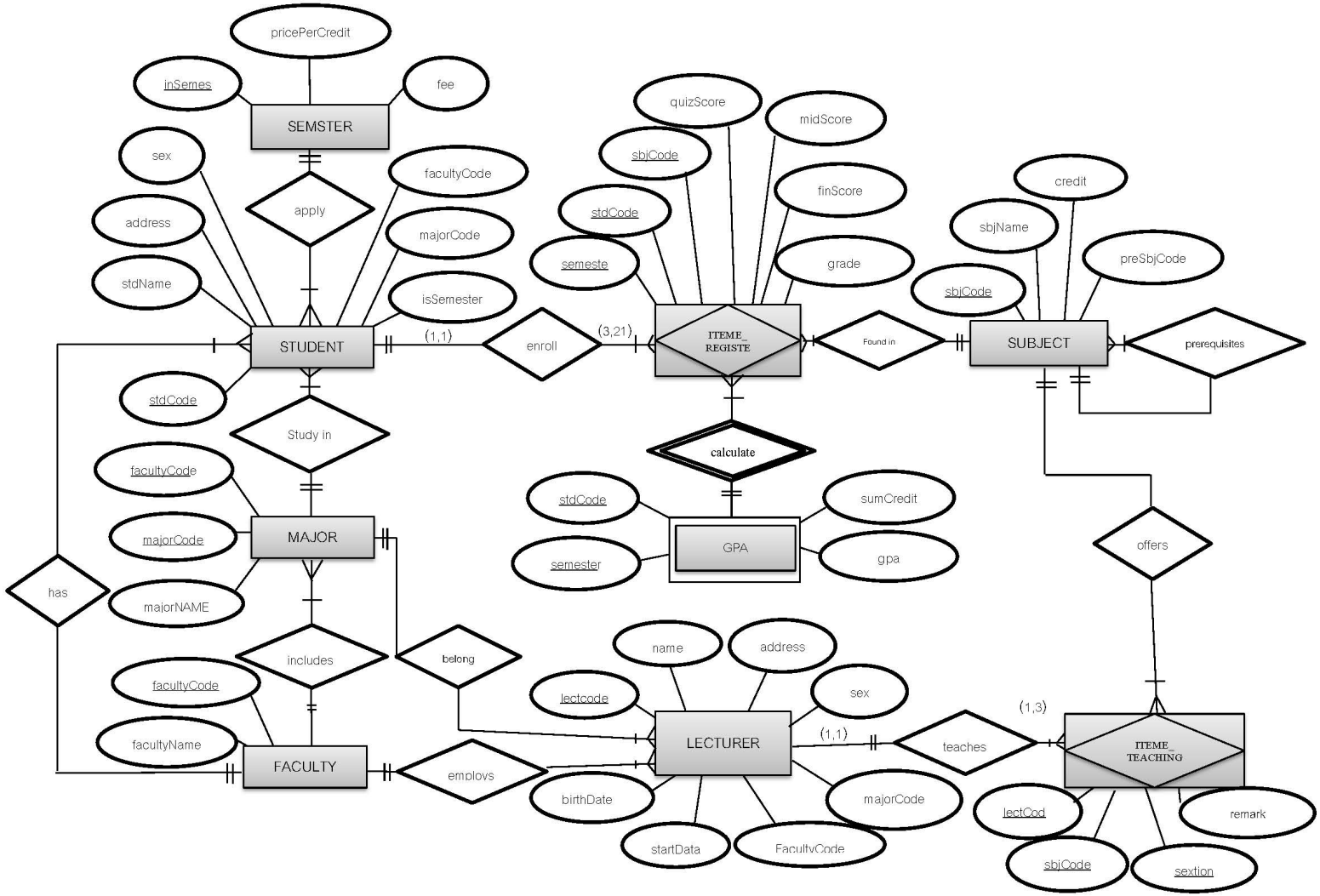
ภาพที่ 5.48 เป็นแบบจำลอง E-R ฉบับสมบูรณ์ของระบบงานทะเบียนและวัดผลของวิทยาลัยแห่งหนึ่งโดยจะพบว่าได้เพิ่ม weak entity ชื่อ GPA ขึ้นมาอีกหนึ่งเอนทิตี ซึ่งใช้สำหรับเก็บผลการเรียนหรือเกรดเฉลี่ยของนักศึกษาในแต่ละภาค และในภาพที่ 5.49 จะเป็นแบบจำลอง E-R ที่สร้างขึ้นจากโปรแกรม Microsoft Visio โดยเป็นแบบจำลองในรูปของ Crow's Foot Model



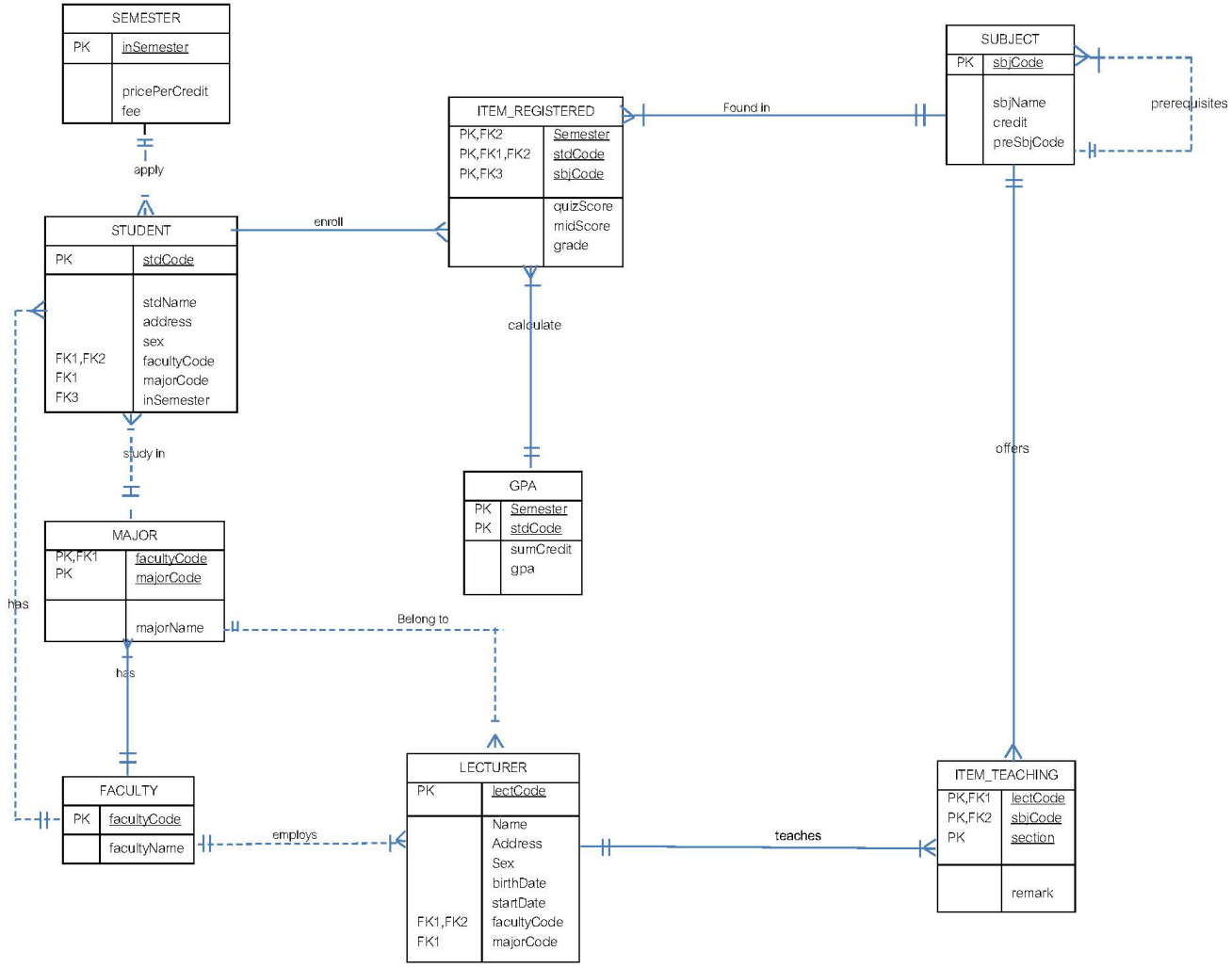
-คณะวิชา <*ว่าจ้าง*> อาจารย์สอนได้หลายคน

-อาจารย์ผู้สอนหลายคน <*สังกัด*> อยู่ในสาขาวิชาของตน

ภาพที่ 5.47 แบบจำลอง E-R แสดงความสัมพันธ์แบบ 1:M ระหว่างเอนทิตี FACULTY และ LECTURER



ภาพที่ 5.48 แบบจำลอง E-R ระบบงานทะเบียนฉบับสมบูรณ์ของวิทยาลัยแห่งหนึ่ง



ภาพที่ 5.49 แบบจำลอง E-R ฉบับสมบูรณ์ ของระบบงานทะเบียนและวัดผล ในรูปแบบ Crow's Foot Model

สรุป

แบบจำลอง E-R เป็นแบบจำลองเชิงแนวคิดที่นำเสนอด้วยภาพหรือไดอะแกรม ซึ่งใช้สื่อสารให้เกิดความเข้าใจระหว่างผู้วิเคราะห์กับผู้ใช้ แบบจำลอง E-R จะนำเสนอรายละเอียดหรือข้อมูลต่างๆ ของธุรกิจในภาพรวมว่ามีเอนทิตีและแอตทริบิวต์อะไรบ้าง มีการเชื่อมโยงความสัมพันธ์ระหว่างเอนทิตีอย่างไร ความสัมพันธ์แบบยูนารี เป็นความสัมพันธ์ระหว่างเอนทิตีเดียว หรือเรียกว่าแบบรีเคอร์ซีพ ความสัมพันธ์แบบไบนารีจะมีเอนทิตีที่เกี่ยวข้องกัน 2 เอนทิตี ความสัมพันธ์แบบเทอร์นารี เป็นความสัมพันธ์ที่มีเอนทิตีเข้ามาเกี่ยวข้อง 3 เอนทิตีด้วยกัน กฎเกณฑ์ทางธุรกิจ คือถ้อยแถลงหรือถ้อยคำที่กำหนดขึ้นมา ซึ่งเกี่ยวกับกฎเกณฑ์ข้อบังคับทางธุรกิจ โดยมีจุดมุ่งหมายเพื่อรักษา ควบคุม และบังคับใช้ให้เป็นไปตามโครงสร้างและพฤติกรรมในธุรกิจนั้นๆ กฎเกณฑ์ทางธุรกิจ สามารถนำไปปฏิบัติตามด้วยการผ่านกระบวนการทำงานของพนักงานที่ปฏิบัติหน้าที่อยู่เป็นประจำ หรืออาจถูกกำหนดไว้ในโปรแกรมประยุกต์ก็เป็นได้ ข้อบังคับบนความสัมพันธ์ มีอยู่ 2 ชนิดคือ Cardinality Constraints และ Participation Constraints ปัญหาที่เกิดขึ้นในแบบจำลอง E-R ประกอบด้วย Fan Traps และ Chasm Traps

แบบฝึกหัดทบทวน

1. แบบจำลอง E-R คืออะไร จงอธิบาย
2. เอนทิตีปกติและเอนทิตีอ่อนแอ มีความแตกต่างกันอย่างไร และเขียนด้วยสัญลักษณ์ใด
3. จงอธิบายความหมายของแอตทริบิวต์โดเมน
4. จงอธิบายความหมายของแอตทริบิวต์ต่อไปนี้ พร้อมยกตัวอย่างประกอบพร้อมสัญลักษณ์
    4.1 แอตทริบิวต์อย่างง่าย
    4.2 แอตทริบิวต์แบ่งย่อย
    4.3 แอตทริบิวต์ที่มีค่าเดียว
    4.4 แอตทริบิวต์ที่มีหลายค่า
    4.5 แอตทริบิวต์ไม่ต้องจัดเก็บ
5. จงอธิบายความสัมพันธ์ต่อไปนี้ พร้อมยกตัวอย่างประกอบ
    5.1 ความสัมพันธ์แบบยูนารี
    5.2 ความสัมพันธ์แบบไบนารี
    5.3 ความสัมพันธ์แบบเทอนารี
6. จงอธิบายความสัมพันธ์ระหว่างเอนทิตีต่อไปนี้ พร้อมยกตัวอย่างประกอบ
    6.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง
    6.2 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม
    6.3 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม
7. หัวหน้าสาขาหนึ่งคนดูแลสาขา 1 สาขา สามารถเขียนเป็นแบบจำลอง E-R ได้อย่างไร
8. พนักงานขายหนึ่งคนสามารถดูแลลูกค้าได้หลายคน สามารถเขียนเป็นแบบจำลอง E-R ได้อย่างไร
9. ปัญหา Fan Traps ที่เกิดขึ้นในแบบจำลอง E-R เกิดจากสาเหตุใด แล้วมีวิธีแก้ไขปัญหาอย่างไร
10. ปัญหา Chasm Traps ที่เกิดขึ้นในแบบจำลอง E-R เกิดจากสาเหตุใด แล้วมีวิธีแก้ไขปัญหาอย่างไร
11. จงวิเคราะห์ระบบงานห้องสมุดและนำมาเขียนเป็นแบบจำลอง E-R

เอกสารอ้างอิง

Batini, C. (1992). *Conceptual Database Design: An Entity-Relationship Approach.* USA: The Benjamin/Cummings Publishing.

Chittayasothorn, S. (2007). *Relational Database Design.* Thailand: Institute of Thai Information Technology.

Coronel, C., Morris, S., & Rob, P. (2013). *Database System: Design, Implementation, and Management* (10th ed.). USA : Course Technology.

Eric, R., & Jim, R. W. (2012). *Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement.* USA: The Pragmatic Programmers.

Sikha, B., & Richard, E. (2012). *Database Design Using Entity-Relationship Diagrams* (2nd ed.). UK: Taylor & Francis Ltd.

Teorey, T. J. (2006). *Database Modeling and Design: Logical Design* (4th ed.). USA: Morgan Kaufmann.